



NATIONAL UNIVERSITY OF IRELAND, GALWAY

DOCTORAL THESIS

Schema-Agnostic Queries for
Large-Schema Databases: A
Distributional Semantics Approach

Author:

André FREITAS

Supervisor:

Dr. Edward CURRY

Research Director:

Prof. Dr. Stefan Decker

Examiners:

Dr. Christopher Brewster

Dr. Paul Buitelaar

Dr. John Breslin

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

June 2015

Declaration of Authorship

I, André FREITAS, declare that this thesis titled, 'Schema-Agnostic Queries for Large-Schema Databases: A Distributional Semantics Approach' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

*“Austere line of the far-off coast-
When the ship comes near, the slope raises up
In trees where the Distance had been empty;
Closer by, the land opens up in sounds and colours:
And, on disembarking, there are birds, flowers,
Where, from afar, there had only been a meaningless line.*

*The dream consists in seeing the invisible shapes
Of the hazy distance, and, with perceptible
Movements of hope and will,
Search out in the cold line of the horizon
The tree, the beach, the flower, the bird, the spring-
The well deserved kisses of Truth.”*

Horizon, Fernando Pessoa

Abstract

Insight Centre for Data Analytics
Digital Enterprise Research Institute (DERI)

Doctor of Philosophy

Schema-Agnostic Queries for Large-Schema Databases: A Distributional Semantics Approach

by André FREITAS

The evolution of data environments towards the growth in the *size, complexity, dynamicity* and *decentralisation* (SCoDD) of schemas drastically impacts contemporary data management. The SCoDD trend emerges as a central data management concern in Big Data scenarios, where users and applications have a demand for more complete data, produced by independent data sources, under different semantic assumptions and contexts of use. Most Database Management Systems (DBMSs) today target a *closed communication* scenario, where the symbolic schema of the database is known a priori by the database user, which is able to interpret it in an unambiguous way. The context in which the data is consumed and produced is well-defined and it is typically the same context in which the data was created. In contrast, data management under the SCoDD conditions target an *open communication* scenario where the symbolic system of the database is unknown by the user and multiple interpretation contexts are possible. In this case the database can be created under a different context from the database user. The emergence of this new data environment demands the revisit of the semantic assumptions behind databases and the design of data access mechanisms which can support semantically heterogeneous (open communication) data environments.

This work aims at filling this gap by proposing a complementary semantic model for databases, based on *distributional semantic models*. Distributional semantics provides a complementary perspective to the formal perspective of database semantics, which supports *semantic approximation as a first-class database operation*. Differently from models which describe uncertain and incomplete data or probabilistic databases, *distributional-relational models* focuses on the construction of conceptual approximation approaches for databases, supported by a comprehensive semantic model *automatically built* from large-scale unstructured data external to the database, which serves as a *semantic/commonsense knowledge base*. The semantic model can be used to support *schema-agnostic*

queries, i.e. abstracting the data consumer from a specific conceptualization behind the data.

The proposed distributional-relational semantic model is supported by a *distributional structured vector space model*, named $\tau - Space$, which represents *structured data* under a distributional semantic model representation which, in coordination with a *query planning approach*, supports a schema-agnostic query mechanism for large-schema databases. The query mechanism is materialized in the *Treo query engine* and is evaluated using *schema-agnostic natural language queries*.

The evaluation of the query mechanism confirms that distributional semantics provides a *high-recall, medium-high precision*, and *low maintainability* solution to cope with the abstraction and conceptual-level differences in schema-agnostic queries over large-schema/schema-less open domain datasets. Moreover, the compositional semantic model defined by the query planning mechanism supports *expressive schema-agnostic queries* over large-schema/schema-less open domain datasets. The proposed distributional-relational structured vector space model ($\tau - Space$) materialized as an *inverted index*, supports the development of a schema-agnostic query mechanism with *interactive query execution time*.

Acknowledgements

The process of doing a PhD is usually described as a humbling experience where you are frequently confronted with your limitations and with your status as an apprentice. More importantly, it is a process which forces you to become aware of the army of people that you are dependent on to proceed with the simplest achievement.

I'm eternally in debt to my supervisor Dr. Edward Curry, who accepted me as his apprentice and turned real my main ambition in life: working with science. I'm grateful for this opportunity and it rarely passes a day when I'm not in touch with this gratitude. More importantly, the scientific and personal support given by Edward softened a journey where you are constantly confronted with your insecurities and limitations. I'm also greatly thankful to Dr. Seán O'Riain for accepting me in his team and for his technical and personal support. Seán was my unit leader and co-supervisor during most of the PhD process and did everything to support my period in his unit.

The period I spent in DERI/NUI Galway was a great experience, and I'm happy to be a small part of the history of the research institute. DERI had a unique combination of a hardworking ethics with light-heartedness, and a strong balance between theory and practice. I benefited a lot from the interaction with the institute colleagues and from their kindness. I wish to publicly thank the members of my Graduate Research Committee: prof. Dr. Stefan Decker, prof. Dr. Siegfried Handschuh, Dr. Paul Buitelaar, Dr. John Breslin, Dr. Connor Hayes, Dr. Aidan Hogan, for their constructive feedback. In particular, I would like to thank prof. Dr. Siegfried Handschuh and Dr. Paul Buitelaar for the collaboration opportunities that they offered me in the area of distributional semantics.

Many people contributed and collaborated around the topics that involved this thesis. Prof. Dr. João Carlos Pereira da Silva was a partner in many discussions and projects. João directly helped me to put the concepts in this thesis in a more formal way. Additionally, Chapter 10 contains the work done in collaboration with João. I also thanks the people which were involved in the different phases of the development of the Treo System: João Gabriel Santos (who took part on the development of the first Treo prototype, which I latter fully redeveloped), Fabricio Firmino (who developed most of the Treo interface) and Danilo Carvalho (who optimized the Explicit Semantic Analysis (ESA) framework).

I am also in debt to my evaluators, Dr. Paul Buitelaar and Dr. Christopher Brewster for their feedback, corrections and suggestions.

I was also greatly supported by the DERI administrative staff Claire Browne, Hilda Fitzpatrick, Gerard Conneely, Andrew Ghallagher.

On a personal level, I would like to thank my father Jose Carlos, my mother Adelaide and grandmother Laci for their love, personal sacrifice and constant and concrete examples.

Finally I would like to thank my wife Lauane which with great courage took this aesthetic path together with me. For me she is the constant reminder of the grounding virtues of the spirit: love (*Philia, Eros, Agape*), simplicity, and gratitude.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	v
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 The Shifting Database Landscape: Increasing Data Variety	1
1.2 The Vocabulary Problem for Databases	2
1.3 Querying vs. Searching	4
1.4 Data Heterogeneity	5
1.5 Schema-agnostic Queries: Addressing the Conceptual Model Heterogeneity	6
1.6 Core Requirements for Schema-agnostic Queries	9
1.7 Existing Approaches to Interact with Heterogeneous Databases	10
1.8 Schema-agnostic Queries for Databases: A Distributional Semantics Ap- proach	12
1.8.1 Semantic Matching & Commonsense Knowledge Bases	12
1.8.2 Distributional Semantic Model	13
1.8.3 Distributional Semantic Relatedness	14
1.8.4 τ – <i>Space</i> : A Distributional-Relational Semantic Model	15
1.9 Open Domain vs. Domain Specific Semantic Matching	16
1.10 Hypothesis	17
1.11 Research Methodology	18
1.12 Contributions	19
1.13 Thesis Outline	21
1.14 Associated Publications	23
2 Semantic Heterogeneity & Schema-Agnostic Queries	26
2.1 Introduction	26
2.2 Contemporary Data Management & Semantic Heterogeneity	27

2.2.1	Contemporary Data Management Environments	27
2.2.2	The Growth of Data Variety	29
2.2.3	Schema-less Databases	32
2.3	The Vocabulary Problem & Schema-agnostic Queries	32
2.3.1	The Vocabulary Problem for Databases	32
2.3.2	Schema-agnostic Queries	34
2.3.3	Querying Semantically Heterogeneous Data	35
2.4	Data Models	36
2.4.1	Introduction	36
2.4.2	Relational Databases	37
2.4.3	Semantic Web Databases & Linked Data	39
2.4.4	The RDF(S) Data Model	42
2.4.5	Entity-Attribute-Value(EAV/Classes & Relations (CR))	43
2.4.6	Data Model Mappings	44
2.5	Semantic Heterogeneity	44
2.5.1	Introduction	44
2.5.2	Intrinsic Causes of Semantic Heterogeneity	45
2.6	Dimensions of Query-Database Semantic Heterogeneity	46
2.7	Semantic Tractability	49
2.7.1	Basic Concepts	49
2.7.2	Semantic Tractability	51
2.8	Matching Schema-Agnostic Queries	51
2.8.1	Semantic Resolvability	51
2.8.2	Semantic Mapping Types	53
2.8.3	Discussion	57
2.9	Chapter Summary	57
3	Literature Review	58
3.1	Introduction	58
3.2	Requirements	59
3.3	Approaches for Querying Semantic Web/Linked Data Datasets	61
3.4	Entity Search: Vector Space Models for Semantic Web/Linked Data Datasets	62
3.4.1	Sindice/SIREn (Tummarello et al. [94])	62
3.4.2	SWSE (Harth et al. [33])	64
3.4.3	Semplore (Wang et al. [31])	64
3.4.3.1	Dong & Halevy [103]	65
3.4.3.2	SPARK (Zhou et al. [104])	66
3.5	Approximate Queries for Semantic Web/Linked Data Datasets	68
3.5.1	Oren et al. [105]	68
3.5.2	Stuckenschmidt & van Harmelen [35]	68
3.5.3	Hurtado et al. [36]	69
3.5.4	SPARQLer (Kochut et al. [108])	70
3.5.5	iSPARQL (Kiefer et al. [37])	70
3.6	Natural Language Interfaces for Semantic Web/Linked Data Datasets	71
3.6.1	NLP-Reduce (Kaufmann et al. [110])	71
3.6.2	Querix (Kaufmann et al. [111])	72
3.6.3	Ginseng (Bernstein et al. [113])	74

3.6.4	PowerAqua (Lopez et al. [117])	75
3.6.5	Freya [122]	76
3.6.6	ORAKEL [123] & Pythia [43]	77
3.6.6.1	TBSL [44]	78
3.6.7	Question Answering Systems	79
3.7	Visual Query Interfaces for Semantic Web/Linked Data Datasets	80
3.7.1	Semantic Crystal (Sprenger et al [38])	80
3.7.2	QUICK (Zenz et al. [134])	81
3.8	Analysis & Gap Identification	82
3.8.1	Automatic Query Expansion Strategies	89
3.8.1.1	Approaches for Automatic Query Expansion	89
3.8.1.2	Linguistic Methods	89
3.8.1.3	Corpus-specific statistical approaches	90
3.8.1.4	Query-specific statistical approaches	90
3.8.1.5	Search log analysis	91
3.8.1.6	Web data	91
3.8.1.7	Applications of Automatic Query Expansion techniques for structured data	91
3.9	Chapter Summary	92
4	Towards a New Semantic Model for Databases	93
4.1	Introduction	93
4.2	A Semiotic Model for Databases	94
4.2.1	Semiotics	94
4.2.2	Semiotics for Databases	95
4.2.3	Symbolic Grounding in Databases	99
4.2.4	Data Model	101
4.2.5	Conceptual Model	103
4.2.6	Semantic Web, Linked Data Web & Schema-agnostic Queries	103
4.3	Semantic Model for Databases	104
4.3.1	Motivation	104
4.3.2	Semantics: the Epistemological, Formal & Praxis perspectives	104
4.3.3	Requirements for a Semantic Model for Schema-agnostic Queries	105
4.4	Semantic Models	107
4.4.1	The Formal (Logic) Perspective on Semantics	107
4.4.2	The Cognitive Perspective on Semantics	108
4.4.2.1	Prototypes	108
4.4.2.2	Frames	110
4.4.3	The Structuralist Perspective on Semantics	111
4.4.4	Requirements Coverage	113
4.5	Distributional Semantics	114
4.5.1	Introduction	114
4.5.2	Distributional Semantic Models (DSMs)	114
4.5.2.1	Distributional Vector Space	114
4.5.2.2	Context Patterns	116
4.5.2.3	Weighting Functions	116
4.5.2.4	Distributional Matrix	118

4.5.2.5	Dimensionality Reduction	119
4.5.2.6	Distance Measures: Semantic Similarity & Relatedness	119
4.5.2.7	Multiple Senses & Ambiguity	120
4.5.2.8	Distributional Semantic Model	121
4.6	Effectiveness of Distributional Semantics: Semantic Similarity & Relatedness	122
4.6.1	Motivation	122
4.7	A Distributional Semantic Model for Databases	124
4.7.1	Distributional Grounding of Database Symbols	124
4.7.2	Semantic Best-effort	128
4.8	A Semantic Abstraction Layer for Databases	130
4.9	Chapter Summary	132
5	The Semantic Matching Problem: An Information-Theoretical Approach	133
5.1	Introduction	133
5.2	Semantic Matching Model	134
5.3	Semantic Complexity & Entropy	135
5.4	Measures of Semantic Entropy	137
5.4.1	Syntactic Entropy (H_{syntax})	137
5.4.2	Structural Entropy (H_{struct})	139
5.4.3	Terminological Entropy (H_{term})	140
5.4.3.1	Uniform Distribution	140
5.4.3.2	Reference Distribution	141
5.4.3.3	Matching Entropy ($H_{matching}$)	142
5.4.3.4	Uniform Distribution	142
5.4.3.5	Reference Distribution	143
5.4.3.6	Background Knowledge Entropy (H_{div})	143
5.5	Minimizing the Entropy for the Semantic Matching	145
5.5.1	Semantic Pivoting	146
5.5.2	Syntactic Matching	147
5.6	Chapter Summary	147
6	τ – Space: A Hybrid Distributional-Relational Semantic Model	149
6.1	Introduction	149
6.2	RDF(S) Data Model & Semantic Model	150
6.2.1	Motivation	150
6.2.2	Lexical Categories for the Data Model Elements	150
6.2.3	Link Types	152
6.2.4	Factors Affecting Interpretability	156
6.3	Distributional-Relational Model (DRM)	159
6.4	Distributional Structured Semantic Space (τ – Space)	161
6.4.1	Introduction	161
6.4.2	Building the τ – Space	161
6.4.3	Vector Basis	162
6.4.4	Word Space (VS^{word})	163
6.4.5	Distributional Space (VS^{dist})	164

6.4.6	Unification of the Distributional and the Word Spaces	164
6.4.7	Instance subspaces (VS^I)	165
6.4.8	Property subspaces (VS^P)	165
6.4.9	Class subspaces (VS^C)	166
6.4.10	Real dimension (\mathbb{R})	167
6.4.11	Property Relation Vectors	168
6.4.12	Class Relation Vectors	168
6.4.13	Building the τ – <i>Space</i>	169
6.4.14	τ – <i>Space</i> Example	170
6.4.15	Complementary Structures: Reification subspace	171
6.4.16	Compositionality & Semantic Interpretation	174
6.5	Discussion	174
6.5.1	Transportability as Coordinate Transformations	175
6.6	Tensor Representation	176
6.7	The τ – <i>Space</i> as an Inverted Index	176
6.8	Representation of Complex Classes	178
6.8.1	Introduction	178
6.8.2	The Structure of Class Descriptors	179
6.8.3	Representation Model	182
6.8.3.1	Overview	182
6.8.3.2	Representation Elements	182
6.8.4	Extending the τ – <i>Space</i> for Complex Class Descriptors	186
6.9	Chapter Summary	187
7	Distributional Semantic Search	188
7.1	Introduction	188
7.2	Distributional Semantic Models	189
7.2.1	Selecting the Distributional Semantic Models	189
7.2.2	Latent Semantic Analysis (LSA) (Landauer & Dumais, 1997)	189
7.2.3	Random Indexing (RI) (Karlgrén & Salhgren 2001)	190
7.2.4	Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch, 2007)	191
7.3	Evaluating the Suitability of Distributional Semantic Relatedness Measures	192
7.3.1	Overview	192
7.3.2	WordNet-based measures	192
7.3.3	Comparative Analysis	194
7.4	Distributional Semantic Search	196
7.4.1	Motivation	196
7.4.2	Distributional Semantic Relatedness Measure as a Ranking Function	197
7.4.3	Inverted-index Distributional Semantic Search	198
7.4.4	Building the Semantic Space	200
7.4.5	Searching the Semantic Space	202
7.5	Evaluating the Distributional Semantic Search: Searching for Database Predicates	203
7.5.1	Motivation	203
7.5.2	Evaluating the Terminology-level Search	204
7.5.3	Qualitative Analysis	204
7.5.4	Quantitative Analysis	207

7.5.5	Analysis of the Distributional Space Dimensionality	208
7.6	Semantic Differential Analysis	208
7.7	Chapter Summary	212
8	The Schema-agnostic Query Processing Approach	213
8.1	Introduction	213
8.2	Overview of the Schema-Agnostic Query Approach	214
8.3	Principles of the Schema-agnostic Query Approach	215
8.4	SPARQL Semantics	216
8.4.1	Motivation	216
8.4.2	Basic Definitions	216
8.5	Query Analysis	220
8.5.1	Motivation	220
8.5.2	Query Representation	220
8.5.3	Query Analysis Steps	223
8.5.3.1	Overview	223
8.5.3.2	Query Parsing	224
8.5.3.3	Entity Recognition & Classification	225
8.5.3.4	Core Entity Identification	226
8.5.3.5	Query Transformation	228
8.5.4	Query Entity Types & Database Structural Interpretation	229
8.5.5	Query Classification	230
8.6	Schema-agnostic Query Processing	231
8.6.1	Overview	231
8.6.2	Search operations	231
8.6.2.1	Instance search	232
8.6.2.2	Class search	233
8.6.2.3	Property search ($VS^{P\{dist\}}(i)$ & $VS^{P\{dist\}}$)	235
8.6.3	Constraint Composition & Solution Modifiers	235
8.6.3.1	Constraint Composition	235
8.6.3.2	Solution Modifiers	237
8.6.3.3	User Feedback Modifiers	240
8.6.4	Operation Composition & Planning	240
8.6.5	Geometrical Interpretation	240
8.6.6	Query Processing Examples	242
8.6.6.1	Query Example I:	242
8.6.6.2	Query Example II:	246
8.7	The Treo System	247
8.7.1	Overview	247
8.7.2	Architecture	248
8.7.3	User Interface	249
8.7.4	Examples	251
8.7.5	Implementation	255
8.7.6	EasyESA: A Distributional Semantics Infrastructure	256
8.8	Chapter Summary	258
9	Evaluation	259

9.1	Introduction	259
9.2	Evaluation Methodology	260
9.3	Test Collection Analysis	261
9.3.1	Motivation	261
9.3.2	Dataset Analysis	263
9.3.2.1	Dataset semantic size & heterogeneity requirement	263
9.3.3	Query Set Analysis	264
9.3.3.1	Test Collection Format	264
9.3.3.2	Structural Variability	265
9.3.4	Conceptual/Vocabulary Gap Patterns	267
9.3.4.1	Realistic & Representative Query Set: Comparative Analysis with Query Logs	269
9.3.4.2	Dependency of the evaluation on the QALD Dataset	271
9.4	Evaluation components & performance metrics	272
9.5	Evaluation Setup	273
9.6	Relevance	273
9.6.1	Relevance Metrics	273
9.6.1.1	Relevance Results Analysis	275
9.6.2	Query Type Relevant Results	277
9.6.3	Component Relevance Results	278
9.7	Temporal & Index Size Performance Evaluation	280
9.8	Transportability Evaluation	281
9.9	Critical Post-mortem Analysis	282
9.10	Comparative Evaluation with Existing QALD Systems	283
9.11	Requirements Coverage	283
9.12	Chapter Summary	284
10	Generalization & Further Applications	285
10.1	Semantic Approximations at Scale	285
10.2	Knowledge-based Semantic Interpretation	286
10.2.1	Core Principles	286
10.2.2	Semantic Interpretation Model	287
10.2.3	KB-based Grammar	288
10.3	Further Applications: Approximate and Selective Commonsense Reasoning	291
10.3.1	Introduction	291
10.3.2	Motivational Scenario	292
10.3.3	Embedding the Commonsense KB into the τ -Space	293
10.3.4	Distributional Navigation Algorithm	293
10.3.5	Evaluation	295
10.3.5.1	Setup	295
10.3.5.2	Reasoning Selectivity	297
10.3.5.3	Semantic Relevance	298
10.3.5.4	Addressing Information Incompleteness	298
10.3.6	Analysis of the Algorithm Behavior	299
10.4	Further Applications: Distributional Logic Programming	303
10.4.1	Motivation	303
10.4.2	Motivational Scenario	303

10.4.3	Distributional Logic Programs	305
10.5	Related work on the interface between structured data, logics and distributional semantics	307
10.6	Hybrid Distributional-Relational Models (DRMs)	309
10.6.1	Types of Distributional-Relational Models	309
10.6.1.1	Semantic Matching	309
10.6.1.2	Knowledge Discovery	310
10.6.2	The Distributional Data Stack	310
10.7	Chapter Summary	311
11	Conclusion	312
11.1	Thesis Summary	312
11.2	Conclusions	314
11.3	Limitations & Open Questions	320
11.4	Main Contributions	320
11.5	Future Research Directions	321
A	QALD 2011 Query Set	324
B	DBR Semantic Relatedness Gold Standard	343
C	Terminology Search Queries & Results	348
D	Training Query Set	354
E	Relevance Metrics Associated to the Baseline Systems	355
E.1	PowerAqua	355
E.2	Freya	356
	Bibliography	358

List of Figures

1.1	Positioning of different approaches in the expressivity/usability trade-off spectrum. A schema-agnostic query mechanism should be able to have both high usability and expressivity (red dots).	5
1.2	Query expressivity vs query construction time quadrant. Schema-agnostic queries allows both high expressivity and low query construction time. . .	7
1.3	Example of user information need expressed as a natural language query and possible data representations in different conceptual models.	8
1.4	Alignment of the natural language query and dataset entities.	9
1.5	Schematic representation of the main components of the proposed query approach (for the example query in Figure 1.3). Numbers indicate the workflow from the construction of the distributional semantic model to the processing of the query results.	16
2.1	The long tail of data variety.	31
2.2	Data consumption cost associated with schema size.	31
2.3	Visual representation of elements of the relational model.	38
2.4	Layered Semantic Web model.	39
2.5	Datasets available in the Linked Data Cloud circa 2011.	41
2.6	Intrinsic causes of semantic heterogeneity.	46
2.7	Taxonomy of lexico-semantic differences.	49
2.8	Classification of existing queries according to the <i>lexico-semantic differences</i> and <i>semantic mappings</i>	50
2.9	Example of predication differences associated with the database representation derived from a natural language statement.	52
2.10	Semantic resolvability for different mapping types.	56
2.11	Semantic relationships between different semantic mapping configurations for abstraction and context (adapted from Kashyap & Sheth [27]).	56
3.1	A taxonomy of approaches to AQE.	89
4.1	Semiotic triangle (Loebner, 2014 [149]).	95
4.2	Projection of the semiotic triangle to database (George, 2005 and Sheth & Larson, 1990 [87, 88]).	95
4.3	Elements involved in the human-database communication.	98
4.4	Core elements and concepts involved in the human-database communication. Focus on the elements related to the natural language aspects of the user-database communication.	99
4.5	Semiotic triangle for the cognitive, structuralist and formal perspectives of semantics (Loebner, 2014 [149]).	111

4.6	Depiction of the example of the distributional semantic representation of a word in a corpora.	118
4.7	Distributional matrix built from the context vectors of the target words. . .	118
4.8	Depiction of the cosine similarity for the distributional vector space. . . .	120
4.9	Depiction of word sense components for the distributional vector of a word.	121
4.10	Depiction of distributional relations, contexts and different representation views for distributional semantics.	125
4.11	Semantic models of database elements.	125
4.12	Logical (A) versus Distributional (B) alignment between query and database elements.	126
4.13	Depiction of the vector representation for the semantic relatedness threshold.	127
4.14	Example query and predicates.	127
4.15	List of database predicates for the example database ranked by their semantic relatedness score against a query term.	127
4.16	d-alignment between query term and database term.	128
4.17	Corresponding rules.	129
4.18	Distributional semantics layer complementing the database semantics. . .	131
4.19	Semantic completeness of databases.	131
5.1	Abstraction reflecting the process of semantically mapping query to database elements.	135
5.2	Generic steps for the query processing and associated entropy measures for each step.	138
5.3	Instantiation of the query-entropy model for the example query.	145
6.1	Link patterns at the instance-level.	153
6.2	Link patterns at the terminology-level.	153
6.3	Analysis of the interpretability dimensions of databases.	159
6.4	Distributional semantics layer complementing the database semantics. . .	160
6.5	Distributional vector representation for instances, properties and classes. .	167
6.6	Vector representation for the distributional subspaces associated with instances (contextualised).	167
6.7	Property relation vectors in the $\tau - Space$	168
6.8	Class relation vectors in the $tau - Space$	169
6.9	Topological relationship between the vector spaces that generate the $\tau - Space$	170
6.10	Creation of the instance subspaces $VS^I (VS^{word}, VS^{dist})$	172
6.11	Creation of the class subspaces $VS^C (VS^{word}, VS^{dist})$	172
6.12	Creation of the property subspaces $VS^P (VS^{dist})$	172
6.13	Creation of the parametrized subspaces $VS^P(i), VS^C(i) (VS^{dist})$	173
6.14	Creation of the relation vectors.	173
6.15	Reification extension of the core $\tau - Space$	173
6.16	Dimensions of the distributional tensor.	176
6.17	Matrix projections of the $\tau - Space$ tensor.	177
6.18	Distributional inverted index structure.	178
6.19	Excerpt of the Wikipedia category links associated with the ‘Barack Obama’ article.	180

6.20	Long tail distribution of POS Tag sequences for Wikipedia categories.	181
6.21	Graph patterns showing the relations present in the graph representation.	185
6.22	Categories following the representation model.	186
6.23	Depiction of the structure of the complex class subspace.	187
7.1	Distributional semantic search example.	198
7.2	Inverted index representation for the distributional semantic space.	199
7.3	Interfaces for the interaction with the DSMs.	201
7.4	ESA distributional semantic space construction.	201
7.5	Workflow for the construction of the ESA distributional space.	201
7.6	Terminological semantic space search process.	202
7.7	Examples of ESA interpretation vectors for <i>United States Senators from Illinois</i> and <i>spouse</i>	203
7.8	Values for the weights of the context vectors.	203
7.9	Set of example queries over the DBpedia vocabulary and top-8 results.	205
7.10	Additional set of example queries over the DBpedia vocabulary and top-8 results.	206
7.11	Example of the conjunction of two predicates.	207
7.12	Growth in the dimensionality of the space by the distributional semantic model.	209
7.13	Possible nominal classification systems for the semantic relatedness values.	209
7.14	Semantic relatedness scores for sample query-vocabulary matches.	210
7.15	Depiction of the elements of the semantic differential model over a ranked list of results.	211
8.1	High-level workflow of the steps of the query processing approach.	214
8.2	Example of the query analysis output for the query: ‘Who is the daughter of Bill Clinton married to?’.	223
8.3	Example of the query analysis output for the query: ‘What is the highest mountain?’.	223
8.4	Components of the query analysis.	224
8.5	Example of the query POS Tagging and dependency parsing.	225
8.6	Example of the entity recognition for the query.	225
8.7	Example of the entity classification for the query.	226
8.8	Determination of the core entities for the two example queries.	227
8.9	Possible interpretation for the query examples.	229
8.10	Vector representation for the property path composition.	237
8.11	Vector representation for the extensional expansion.	237
8.12	Vector representation for the star-shaped property composition.	238
8.13	Query processing steps for the query example (Part I).	244
8.14	Query processing steps for the query example (Part II).	245
8.15	τ – <i>Space</i> query.	246
8.16	Execution of a query processing plan for the query ‘ <i>What is the highest mountain ?</i> ’ (Part I)	248
8.17	Execution of a query processing plan for the query ‘ <i>What is the highest mountain ?</i> ’ (Part II)	249
8.18	High-level components diagram of the schema-agnostic query approach.	250
8.19	Screenshot of the initial query interface.	251

8.20	Screenshot of the result of the Treo engine for the query <i>'Is Margaret Thatcher a chemist?'</i>	252
8.21	Screenshot of the result of the Treo engine for the query <i>'Who is the daughter of Bill Clinton married to?'</i>	252
8.22	Screenshot of the result of the Treo engine for the query <i>'What is the highest mountain?'</i>	253
8.23	Screenshot of the result of the Treo engine for the query <i>'How tall is Claudia Schiffer?'</i>	253
8.24	Screenshot of the result of the Treo engine for the query <i>'Give me all cities in New Jersey with more than 100000 inhabitants?'</i>	254
8.25	Screenshot of the vocabulary search interface for the query 'geology'.	254
8.26	Screenshot of the vocabulary search interface for the query 'bass'.	255
8.27	Result filtering component.	255
9.1	Question item from the QALD 2011 test collection.	265
10.1	Schematic representation of the knowledge-based semantic interpretation model.	288
10.2	Schematic representation of semantic interpretation model mapping syntactic structures to logical forms.	289
10.3	(1) Selection of meaningful paths, (2) Coping with information incompleteness.	292
10.4	Contextual (selected) paths between battle and war.	300
10.5	# of occurrences for pairwise semantic relatedness values, computed by the navigational algorithm for the test collection (paths of length 2, 3, 4). Semantic relatedness values for nodes from distances 1, 2, 3 from the source: increasing semantic relatedness to the target.	302
10.6	Increasing variation of the semantic relatedness values as navigated nodes approach the target node.	302
10.7	Depiction of a set of distributional program-database alignments.	303
10.8	Derivation for the question <i>'Is the father in law of Bill Clinton's daughter a politician?'</i>	307
10.9	Distributional Data stack.	311

List of Tables

1.1	Mapping of the core requirements to the hypothesis.	18
2.1	Correspondence between the categories of the RDF(S), EAV, Relational and First-order logic data models.	44
3.1	Requirements dimensions and their correspondence on literature.	60
4.1	Correspondence between RDF, Logics, Relational and Part-of-Speech (lexical categories) Patterns.	103
5.1	Classification of entropy measures according to associated features.	144
6.1	Correspondence between RDF, Logics, Relational and lexical categories.	151
6.2	Core feature set and examples of categories with different feature types.	180
6.3	Distribution and examples of classes with different feature types.	180
6.4	Distribution and examples of classes with different POS Tags.	181
7.1	Evaluation of the correlation between semantic similarity and relatedness measures and human correlation using the MC, WS-353 and DBR datasets.	196
7.2	Evaluation metrics for the ESA-based terminology-level semantic search.	208
7.3	Comparative analysis of the number of queries answered in relation to two baselines: (i) string matching (stemming) and (ii) WordNet query expansion.	208
7.4	Measures and distribution for the elements semantic differential analysis.	211
8.1	Graph Patterns & Solution Modifiers	219
8.2	Examples of IDF values a over Wikipedia 2013 corpus. The more specific words have higher IDF values.	228
8.3	Basic type graph patterns.	230
8.4	Types of operators.	230
9.1	Requirements for the test collection and associated evaluation metrics.	262
9.2	Dataset metrics.	264
9.3	Dataset semantic size & heterogeneity requirement coverage.	264
9.4	Statistics for the features of the QALD-DBpedia'2011 query set.	265
9.5	Unique query patterns (QALD 2011).	266
9.6	Unique triple patterns (QALD 2011).	267
9.7	Distribution of vocabulary gap types for each entity types (QALD 2011).	268
9.8	POS Tag matching patterns (categorized by data model types) (QALD 2011).	270

9.9	POS Tag matching patterns (aggregated) (QALD 2011).	270
9.10	Comparative analysis between the features of QALD 2011 and USEWOD query logs.	271
9.11	Requirements for the test collection and their associated coverage.	271
9.12	Requirements and associated evaluation metrics.	272
9.13	Aggregate relevance results for the query results (QALD 2011 train + test).	275
9.14	Aggregate relevance results for the query results (QALD 2011 test set).	275
9.15	Relevance results for the query results (Part I).	276
9.16	Relevance results for the query results (Part II).	277
9.17	Aggregated relevance results for the query results grouped by the presence of query feature.	278
9.18	Evaluation of the query processing mechanism results using natural language queries. Measures are collected for the full query mechanism and its core subcomponents: entity search and property search. The measures are categorized according to the query features.	279
9.19	Temporal and size measures of the distributional semantic index.	280
9.20	Dataset adaptation effort.	281
9.21	Comparison with existing systems for the QALD 2011 test set.	283
9.22	Requirements coverage of the proposed schema-agnostic query approach.	284
10.1	# of clauses per relation frequency.	296
10.2	Top-12 frequent relations in the ConceptNet	296
10.3	Selectivity	297
10.4	Incompleteness level.	299
10.5	Examples of semantically related paths returned by the algorithm (Part I).	300
10.6	Examples of semantically related paths returned by the algorithm (Part II).	301
10.7	Examples of semantically related paths returned by the algorithm.	301
10.8	Semantic relatedness determined by the τ -Space module between the predicates in Q and Π , according to arity.	307

Chapter 1

Introduction

1.1 The Shifting Database Landscape: Increasing Data Variety

The Big Data vision is based on the idea of supporting users and information systems with large-scale and comprehensive data. Data sources based on new platforms such as open data, collaborative Web 2.0 platforms, crowd-sourcing, mobile devices and applications, sensors on the Internet of Things, and information extraction frameworks are drastically changing the landscape on data availability and bringing new opportunities for applications which are able to make use of the new data [1].

However, together with its opportunities, Big Data brings together a set of associated *data management* challenges for coping with datasets under a new scale of size and complexity. The most pressing Big Data challenges are summarized as the 3 Vs which are used as a definition for Big Data: *volume*, *velocity* and *variety* [2]. The first two, *volume* and *velocity*, are associated with the demand to process large volumes of data and focus on algorithmic approaches, software and hardware infrastructures to cope with data processing on the new volume scale of datasets.

The *data variety* dimension is related to the demand to cope with databases under larger, more complex and multiple schemas, originating from different databases, semantically heterogeneous and under different data quality assumptions [3]. The increase in data variety drastically impacts the ability of users to access, process and interpret information, demanding new *data management strategies*.

From the perspective of *data access*, for example, existing mechanisms to *query* structured databases based on *structured query languages* does not scale to large schema sizes [4], with potentially thousands or millions of attributes [5]. At this scale, it is

not feasible for data consumers to depend on the manual selection of concepts in the database schema in order to query the data. In order to extract value out of the data, data consumers, including domain experts, casual users and applications will depend on new mechanisms that will support them to efficiently interact, search and query structured data, abstracting data consumers from the specific conceptual representation of the data. Query mechanisms supporting users in the abstraction of the data representation are called *schema-agnostic* [6] or *vocabulary-independent*. These query mechanisms supports users querying databases without the understanding of its conceptual model (schema).

Schema-agnostic queries are intrinsically dependent on a *semantic matching approach* associated with the query mechanism, which is responsible for the semantic mapping of user query terms to database elements. The understanding of the challenges of schema-agnostic queries, the formulation and evaluation of a schema-agnostic query approach and the investigation of its supporting semantic model are the focus of this work.

The following sections introduce the main motivation and provides an outline to the body of work behind this thesis.

1.2 The Vocabulary Problem for Databases

Big Data brings inherent challenges in the way users and applications consume the available data. Users accessing Big Data on the Web or in an organisational environment should be able to query and search data spread over a potentially large number of semantically heterogeneous and large-schema datasets [4].

In order to query the data using structured queries, users need be aware of the structure and the terms used in the data representation (database schema). In the Big Data scenario, where data is potentially spread across large-schema, multiple and heterogeneous datasets, the *semantic gap* between users and datasets becomes one of the most important issues for data consumers. At this scale, it is not convenient, and sometimes not possible, to become aware of the database schema elements in order to build a structured query under the database conceptual model constraints. The dependency between the construction of structured queries and the schema size (for both single and multiple datasets) limits the use of structured queries for large-schema databases.

There is no quantitative study measuring the dependency between schema-size and the effort necessary to query a database, leaving open the definition at which scale a database becomes large from the user perspective. In the scope of this work, it is understood that a schema is large if it leads to a significant effort in the process of matching the user

vocabulary terms which expresses his information needs to the entities in the database. A mapping is considered a significant effort if the symbolic mapping effort accounts for a large proportion of the query construction effort. This effort is dependent on the database schema-size (size of the symbol space which is the target of the user query) and the familiarity of the user to the sets of elements in the database schema.

In addition to the trend of large-schemas, structured database content and their associated schemas are being built in a decentralized way, with multiple actors collaboratively conceptualising and describing a domain without a central coordination authority [7, 8]. In this scenario, differences in conceptualisations intrinsic to the multiplicity of database designers generate databases which are conceptually more heterogeneous, i.e. manifesting terminological and structural variations (Section 1.4).

From the perspective of data consumers, they should be abstracted away from the representation of the data. The *semantic gap* between *user information needs materialized as queries* and the database *structured data representation* is at the center of this problem.

The human interaction with information systems always depended upon artefacts with well-defined terminologies (vocabularies) and syntax. Command-line interfaces, source code interfaces and database schemas are examples of information systems' artefacts containing an associated lexicon and syntax/structure. The use of any of these categories of artefacts depends upon an a priori understanding of the vocabulary and the syntax employed. Since these representations are based on human language which can be ambiguous, vague, inconsistent and variable, a semantic gap between users and the vocabulary-dependent artefacts is created. This concept was described by Furnas et al. [9] as the *vocabulary problem in human-system communication*, a problem associated to the semantic variability intrinsic to the human language.

Until recently, the *vocabulary problem for databases* have not represented a significant limitation due to the size of database schemas and the number of datasets that were available for a data consumer. As more datasets become available in different domains (for example open and sensor data), there is an increasing demand to provide mechanisms that enable both domain experts and casual users [4, 10] to explore and access structured data.

In order to address the vocabulary problem, different information retrieval techniques based on *automatic query expansion* (Carpineto & Romano [11], Sun et al. 2006 [12], (Qiu and Frei 1993 [13], Bast et al. 2007 [14], Crouch and Yang 1992 [15], Schuetze and Pedersen 1997 [16], in Gauch et al. 1999 [17], Hu et al. 2006 [18], Park and Ramamohanarao 2007 [19], Milne et al. 2007 [20], Lam-Adesina & Jones 2001 [21], Chang et al. 2006 [22], Xu & Croft 1996 [23], Kraft & Zien [24]) and *query formulation*

(Zhang et al. 1999 [25] and Goldman et al. 1997 [26]) were proposed. Most of these approaches have been applied in the context of keyword-based information retrieval and have been limited in addressing the vocabulary problem for databases.

1.3 Querying vs. Searching

The availability of a query mechanism for structured data which supports data consumers with *expressive queries* (queries which are able to make use of the conceptual structure behind the database and of the supported database operations) at the same time it abstracts them away from the representation (i.e. being schema-agnostic) is still an active research challenge.

The simplicity and intuitiveness of search engine interfaces, where users search the Web using keyword queries, was a key element in the widespread adoption of search engines for the Web of Documents and in the process of maximizing the value of the information available on the Web. On the other side of the spectrum, from the perspective of structured/semi-structured data consumption, users expect with precise and expressive queries. In this scenario, most users query data with the help of structured queries such as SQL¹ or SPARQL². In the Big Data scenario structured query approaches do not completely address all search and query usability requirements from all categories of users (such as being accessible to casual users and supporting lower query construction times for expert users).

With the Web users have recognized search to be a first-class activity. The search paradigm used in the Web of Documents, however, cannot be directly transported for querying structured data. Keyword search over data does not provide the desired *expressivity*, while traditional structured query mechanisms have poor *usability*. Query expressivity and usability are two dimensions of database querying which define a trade-off behaviour. Different categories of query/search approaches have emerged, targeting the trade-off between usability and expressivity (Figure 1.1, 1.2) and have achieved some level of success: however they do not fully provide *schema-agnosticism*.

The practical relevance of a schema-agnostic query mechanism lies on the fact that structured data is a fundamental component of data and information system environments and the effort associated with accessing this structured data is still large and heavily mediated by the need of Information Technology experts. Additionally, this effort grows with the growth of schemas and volume of databases.

¹<http://en.wikipedia.org/wiki/SQL>

²<http://en.wikipedia.org/wiki/SPARQL>

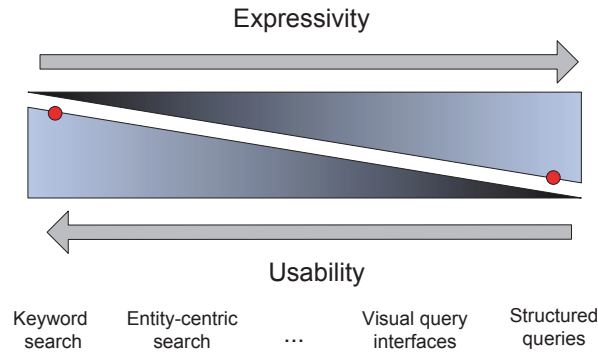


FIGURE 1.1: Positioning of different approaches in the expressivity/usability trade-off spectrum. A schema-agnostic query mechanism should be able to have both high usability and expressivity (red dots).

The dissolution of this trade-off is intrinsically dependent on providing a *semantic matching* approach which enables the alignment or semantic mapping of the data consumer's query to the database conceptual model elements.

1.4 Data Heterogeneity

The vocabulary problem for databases is a consequence of *data heterogeneity* [27], i.e. the multiple realizations in which data can be represented. Even under the same task, different database designers can materialize the same domain into a database using different lexical expressions, conceptualizations, data models, data formats or record granularities [27]. This intrinsic variability in the construction of a database defines an *intrinsic data heterogeneity level* between different databases.

Similarly, there is an intrinsic heterogeneity between a specific database representation and the data consumers mental representation of a domain. If asked to materialize their information needs as free queries (e.g. using natural language) data consumers would be likely to use different terms and structures in the query formulation, a fact which is supported by [9]. The intrinsic heterogeneity is mediated by the role of phenomena intrinsic to natural language such as *synonymy*, *ambiguity* and *vagueness*.

Data heterogeneity becomes a more present concern as users start to query data from different datasets, which were built by independent parties [28]. In this scenario, one starts to move from a *centralized* schema and data model scenario, where data is integrated under a single representation model, to a *decentralized* scenario where data from different schemas and data models are brought together into a different data consumption context [4, 8].

The concept of data heterogeneity encompasses different dimensions:

1. *Conceptual model heterogeneity*: Different domains can be conceptualized using different abstractions and lexical expressions, which are dependent on the intended use behind the database and on the background of the individuals modeling the domain. Given a modeling task with a minimum level of complexity, it is unlikely that two independent parties will generate identical conceptual models [9, 27]. Semantic heterogeneity emerges as a central concern in the Big Data scenario when, data from multiple datasets, developed by different third-parties, need to be accessed and processed in a different context. Conceptual model heterogeneity includes distinct classes of differences which define the conceptual gap. These different dimensions of semantic heterogeneity are further investigated in Section 1.5.
2. *Format heterogeneity*: Covers different formatting assumptions for values. This dimension covers *notational* and *measurement units* differences. Examples of value types dependent on data format are *currency*, *numerical values* and *date-time values*. *Abbreviations* and *acronyms* are also included in this category.
3. *Data model heterogeneity*: Data models provide the syntactical model in which different data objects are represented. Different data sources can be represented using different data models. Examples of data models include the relational model, Resource Description Framework (RDF), eXtensible Markup Language (XML), among others.

The three data heterogeneity dimensions are orthogonal and impact the reconciliation of model dimensions between different databases and the ability of users to query a database. The larger the gap between two models (data, format or conceptual), the larger is the cost of querying or data integration.

This work concentrates on the investigation of the dimension of *conceptual model heterogeneity* and on the ability to automatically bridge the gap between the user and database conceptual models.

1.5 Schema-agnostic Queries: Addressing the Conceptual Model Heterogeneity

The abstraction of users from the database conceptual model is intrinsically connected with the provision of a principled semantic matching mechanism to cross the conceptual gap between the user query and the data representation. Query mechanisms which

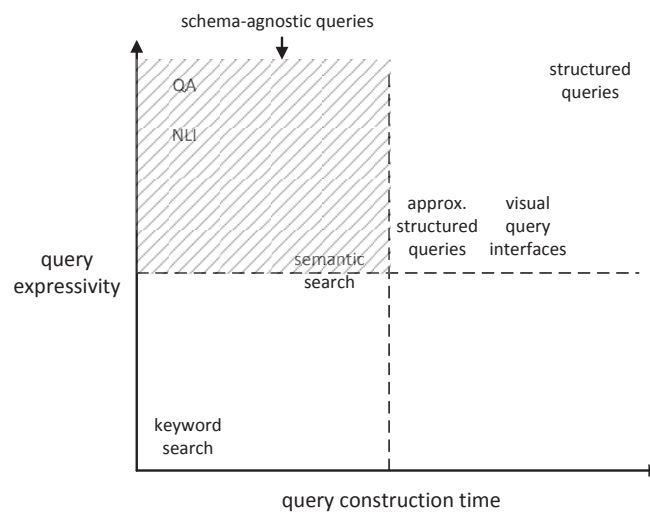


FIGURE 1.2: Query expressivity vs query construction time quadrant. Schema-agnostic queries allows both high expressivity and low query construction time.

are able to provide this abstraction mechanism are described as *schema-agnostic* or *vocabulary-independent queries*. A motivational scenario example is introduced below.

Suppose a user has an information need expressed as the natural language query ‘*Who is the daughter of Bill Clinton married to?*’ (Figure 1.3). The person has access to different databases which contain data that can help addressing the information need. However, the data representations inside the target databases do not match the *vocabulary* and *structure* of the natural language query.

Figure 1.3 depicts an example of the semantic gap between the example user query and possible representations for the data for triples supporting answers for the query. In (a) ‘*daughter*’ and ‘*married to*’ in the query maps to ‘*child*’ and ‘*spouse*’ in the data, in (b) these query terms map to ‘*child*’ and ‘*father of*’ respectively while in (c) the query information related to ‘*daughter*’ is given by the predicate ‘*numberOfKids*’ representing an aggregation in (c), not fully mapping to the query information need.

In order to address query-data alignments, it is necessary to provide a query mechanism which is able to support a semantic matching which copes with the semantic gap between the user query and the data representation. Figure 1.4 shows the alignment between the example natural language query and one possible conceptual model realization. The formulation of a schema-agnostic approach which automatically maps query terms to dataset elements is the core goal of this thesis. A high-level overview of the schema-agnostic query processing steps is depicted in Figure 1.5.



Information Need: Who is the daughter of Bill Clinton married to ?



Semantic Gap

Possible Data Representations

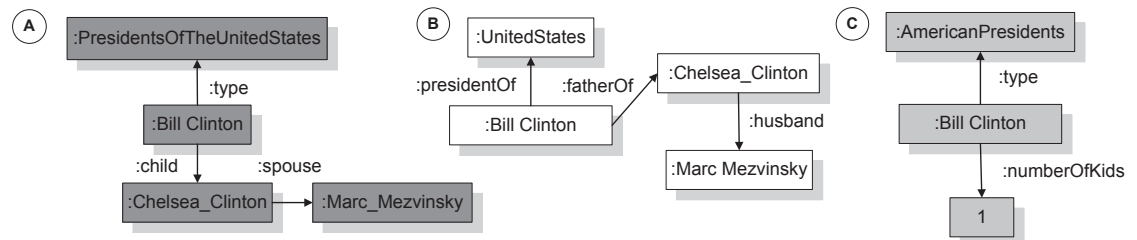


FIGURE 1.3: Example of user information need expressed as a natural language query and possible data representations in different conceptual models.

The semantic gaps previously exemplified, show some dimensions of the semantic differences involved in addressing the vocabulary problem for databases. The core semantic differences can be categorized into the following *high-level dimensions of semantic heterogeneity*:

1. *Abstraction-level differences*: Taxonomical differences between the database representation and the abstraction used in the query. ‘*PresidentOfTheUnitedStates*’ and ‘*AmericanPoliticians*’ express two different sets where the former set is contained in the second. In some cases the abstraction-level expressed in the query may be different from the dataset and a semantic approximation in the abstraction level may return a *semantic best-effort* result, i.e. a (most similar) semantic approximation considering the abstraction-level available in the database. For example, the concept ‘*husband*’ is a gender specific specialization of the ‘*spouse*’.
2. *Conceptual differences*: Consists in different concepts with strongly related/associated meanings in the context of the query, which are not covered by a taxonomical relation. For example the query term ‘*married to*’ maps to the predicate ‘*spouse*’ in the database.
3. *Composition/predication differences*: Information may be expressed under different predicate-argument structures or syntactic compositions. In 1.3(a) ‘*PresidentsOfTheUnitedStates*’ is expressed in a single class, while in 1.3(c) it is expressed as the composition of the predicate ‘*president*’ and its associated object ‘*UnitedStates*’.

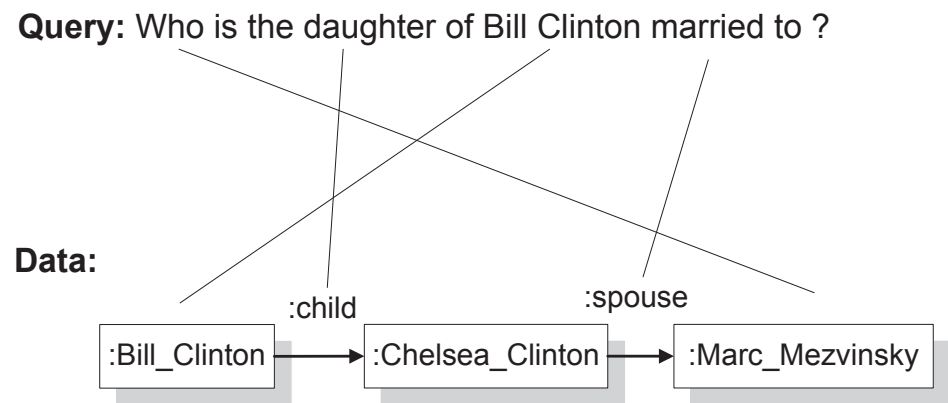


FIGURE 1.4: Alignment of the natural language query and dataset entities.

4. *Operational/functional differences:* Operations such as aggregations and logical filters are a fundamental part of database query expressivity. In a schema-agnostic query scenario, operations can have different lexical expressions (for example ‘*highest*’, ‘*tallest*’ may map to ‘*top most*’ as the database operator). Additionally, operators may implicitly define its association to a specific predicate (predicate selection). For example in the query ‘*What is the highest mountain?*’ the operator ‘*highest*’ may be associated with specific elements in the dataset (e.g. ‘*height*’, ‘*elevation*’).

The semantic heterogeneity dimensions are at the center of the *vocabulary problem for databases* and a *schema-agnostic query mechanism* should be able to address all the dimensions. The process of creating a schema-agnostic query mechanism can be interpreted as the process of performing a *semantic matching* between the user query and the related database entities, attributes, relationships and values. While in a structured query language the semantic matching is manually performed by the users, schema-agnostic queries target the creation principled algorithmic approaches to cope with the dimensions of semantic heterogeneity.

1.6 Core Requirements for Schema-agnostic Queries

The dimensions of semantic heterogeneity are at the center of the schema-agnostic queries, and addressing them can be used to define the *semantic matching requirements* to provide a robust schema-agnostic query mechanism. However, in addition to the requirements related to the semantic matching, schema-agnostic queries need to satisfy other requirements, which are common across search and query mechanisms. A more

in-depth discussion on the determination of the requirements is provided in Chapter 3. The set of requirements are used as qualitative dimensions to evaluate the effectiveness of a schema-agnostic approach.

1. *High usability & Low query construction time:* Support for a simple and intuitive interface for experts and casual users.
2. *High expressivity:* Queries referencing *structural elements and constraints* in the dataset (relationships, paths) should be supported, as well as *operations* over the data (e.g. aggregations, conditions).
3. *Accurate & comprehensive semantic matching:* Ability to provide a principled semantic matching addressing all the dimensions of the semantic heterogeneity problem (abstraction, conceptual, compositional, functional). Semantic matching with high precision and recall.
4. *Low setup & maintenance effort:* Easily transportable across datasets without significant manual adaptation effort. The query mechanism should be able to work under an open domain and across multiple domains. Databases should be indexed with a minimum level of manual adaptations. Minimization of user intervention in the construction of supporting semantic resources used in the semantic matching.
5. *Interactive search & Low query-execution time:* Minimization of user interaction/feedback effort in the query process. Users should get answers with interactive response times³ for most of the queries.
6. *High scalability:* The query approach should scale to large datasets both in query execution and indexing construction time.

1.7 Existing Approaches to Interact with Heterogeneous Databases

There are different strategies for querying databases with varying degrees of flexibility, usability and semantic matching. This work categorizes existing approaches into six categories:

- *Structured queries for databases:* The most traditional way to query databases where users need to explicitly refer to entities, attributes, values, relationships in

³an interactive query execution time is contrasted with a batch query execution time (seconds vs. minutes)

the same vocabulary of the database and should follow the syntax of a structured query language such as SPARQL [29] or SQL.

- *Keyword search for databases:* Focus on the adaptation of keyword-based approaches used in information retrieval for databases. The additional level of flexibility comes from allowing text search over all fields in the database. Existing approaches can vary from bag-of-word approaches where the structural relations in the data are not taken into account (higher usability, lower expressivity) to hybrid keyword-structure queries approaches, where users explicitly reference schema-level information which is used together with the keyword search functionality (lower usability, higher expressivity). Examples of these approaches are available in: [30].
- *Entity search for databases:* Consists of the use of complementary semantic resources to provide additional schema/vocabulary flexibility in the search process. Techniques can vary from: (i) using semantic information (mainly taxonomic relations) in the ontology/vocabulary behind the database, (ii) using external ontological resources, (iii) using linguistic resources, such as WordNet. These approaches include techniques to use the semantic resources effectively in the search process. Semantic search approaches focuses on providing an additional level of vocabulary flexibility at the cost of the construction of the semantic resources. Examples of these approaches are available in: [31, 32, 33, 34]
- *Approximate queries for databases:* Consist of approaches which provide an additional level of flexibility based on the relaxation of structural constraints in the query and in the database. These relaxation operations are usually explicitly defined by users as structured queries operators and parameters. Examples of these approaches are available in: [35, 36, 37]
- *Visual query interfaces for databases:* Query mechanisms which allow users to specify queries or progressively filter query results or navigate through the database with the help of visual elements in the interface. The approaches in this category focus on addressing the semantic gap problem from the perspective of user interaction. Examples of these approaches are available in: [38, 39, 40].
- *Natural language interface (NLI) and question answering (QA) for databases:* NLI focuses on approaches which use *open* or *controlled natural language queries* for querying databases. Results from natural language interfaces may vary from database records to post-processed direct natural language answers (QA). Open (non-controlled) natural language queries are by definition schema-agnostic query mechanisms. Most approaches use a combination of linguistic and ontological

resources to address the query-dataset vocabulary gap. Examples of these approaches are available in: [41, 42, 43, 44, 45].

A detailed survey and gap analysis of each category of the existing approaches are covered in Chapter 3.

Research on natural language interfaces (NLIs) and question answering (QA) systems over structured data have been targeting query scenarios which are schema-agnostic, concentrating on the proposal and evaluation of approaches which put a stronger emphasis on the query-database semantic matching approach. Despite not using the term schema-agnostic, the evaluation of the ability of a query mechanism to semantically match natural language queries to database elements which are unknown by querying agent (user), is a basic premise of the usage scenario of these systems. By explicitly targeting the problem of *schema-agnostic* queries, this work aims at *individuating this concept as a query capability*, emphasizing the importance of this feature in modern data management scenarios and motivating the transference of this capability into other systems. Compared to other research areas such as keyword search over databases or semantic/entity search, NLI/QAs have focused on the development of community-built efforts to evaluate schema-agnostic capabilities under a heterogeneous data scenario (large and conceptually vast databases). Due to the maturity of the test collections and evaluation campaigns in this area this thesis grounds its evaluation on the NLI/QA scenario.

1.8 Schema-agnostic Queries for Databases: A Distributional Semantics Approach

This work focuses on the definition of a *semantic model* to provide a *schema-agnostic query mechanism* to address the *dimensions of the semantic heterogeneity* and the *requirements for schema-agnostic queries* for databases. At the core of the proposed semantic model is the definition of a *distributional semantic model*. This section briefly introduces the core principles and elements of the proposed approach.

1.8.1 Semantic Matching & Commonsense Knowledge Bases

A comprehensive open and multi-domain semantic matching mechanism largely depends on the availability of *large-scale semantic and commonsense knowledge resources* following a principled representation which can be algorithmically processed and used for inference. However, the automatic construction, representation of large commonsense

knowledge bases (KBs) and the provision of associated reasoning mechanisms are still major research challenges which need to be addressed [46, 47, 48]. Knowledge representation and reasoning approaches for commonsense KBs need to cope with the *performance*, *inconsistency* and *incompleteness* problems involved in large-scale reasoning. Another important problem in this area is the *acquisition of commonsense KBs*, i.e. the construction of large-scale commonsense KBs under a specific structured representation scheme [46, 47, 48].

Distributional semantics, combined with a *compositional* model, can provide a principled semantic matching mechanism where, instead of a structured commonsense KB and associated reasoning algorithms, a quantitative statistical semantic model is automatically built from unstructured large-scale corpora. The distributional semantic model addresses the acquisition problem and the representation problem at the expense of some level of inaccuracy, defining an approximative but comprehensive semantic model which can be used to support the semantic query-database matching. The next subsections describe the basic principles of *distributional semantics* and how the schema-agnostic query model is built from it. An in-depth analysis of distributional semantics is provided in Chapter 4.

1.8.2 Distributional Semantic Model

Distributional semantics is built upon the assumption that the context surrounding a given word in a text provides important information about its meaning [49, 50]. A rephrasing of the *distributional hypothesis* states that words that co-occur in *similar contexts* tend to have *similar/related meanings*.

Distributional semantics focuses on the construction of a semantic representation of a word based on the statistical distribution of word co-occurrence in large-scale text collections. The availability of high volume and comprehensive Web corpora brought distributional semantic models as a promising approach to build and represent meaning. Distributional semantic models are naturally represented by *Vector Space Models* (VSMs), where the meaning of a word is represented by a *weighted linguistic context vector*.

However, the proper use of the simplified model of meaning provided by distributional semantics implies understanding its characteristics and limitations. In distributional semantics, *differences of meaning* are mediated by *differences of distribution* in a reference corpora. As a consequence, distributional semantic models allow the *quantification* of the amount of associations or differences in meaning between words. This can be used to quantify the *semantic relatedness* between words. The intuition behind this approach is

that two terms which are highly semantically related in a distributional model are likely to co-occur in similar contexts in the corpora. This allows the automatic construction of a large-scale base of associations from unstructured texts, avoiding the problem of manually building large-scale semantic and commonsense structured KBs.

1.8.3 Distributional Semantic Relatedness

The concept of *semantic relatedness* is described as a generalization of *semantic similarity* [51], where semantic similarity is associated with taxonomic relations between concepts (e.g. *car* and *airplane* share *vehicle* as a common taxonomic ancestor) and semantic relatedness covers a broader range of semantic relations (e.g. *car* and *driver*). Since differences in conceptual models can transcend *taxonomical* differences, the more generic concept of semantic relatedness is more suitable to the query-dataset semantic matching.

Until recently, resources such as WordNet were used in the computation of semantic similarity and relatedness measures. The limitations of the representation present in WordNet include the lack of a rich representation of non-taxonomic relations (fundamental for the computation of semantic relatedness measures) and a limited number of modelled concepts. The effectiveness of statistical models of language [52] and the availability of large amounts of unstructured text on the Web motivated the creation of semantic relatedness measures based on large text collections using distributional semantic models. *Distributional semantic relatedness measures* focus on addressing the limitations of resource-based approaches by trading structure for volume of commonsense knowledge [53].

Comparative evaluations between WordNet-based and distributional approaches for the computation of semantic relatedness measures have shown the strength of the distributional semantics based approaches, reaching a higher correlation level with human assessments [53].

This work uses distributional semantic models (DSMs) and *distributional semantic relatedness measures* as the core mechanism to cope with the *semantic heterogeneity* between user queries and database entities. Since the distributional model is built from large-scale corpora (with a large number of implicit associations) it potentially supports a *comprehensive (high recall)* semantic matching mechanism; as DSMs are automatically built, providing a *low set-up & maintenance effort*.

Moreover, the context in which the distributional semantic model is used for semantic approximation takes into account a reduction of the configuration space in which the set

of candidate query-dataset alignments are defined, using the concept of a *semantic pivot*, which is defined at Chapter 5 and Chapter 8. The semantic pivoting strategy minimizes the uncertainty associated with the distributional semantic model approximation.

1.8.4 $\tau - Space$: A Distributional-Relational Semantic Model

In order to support the demand for a query-database semantic matching using distributional semantics, this work introduces a *hybrid distributional-relational knowledge representation model*, named $\tau - Space$, which adds to the database semantics, the distributional semantics representation. In this work the word ‘relational’ is used in the more generic sense of structured data instead of ‘relational databases’.

The $\tau - Space$ is a *distributional structured vector space model* which represents *structured data graphs* under a distributional semantic model representation. The co-occurrence associational information extracted from an independent *reference corpora* defines a *distributional vector space* where the *labels* associated with constants or predicates of the dataset can be resolved into a geometrical vector of the distributional vector space. The *structured data graph* represents an arbitrary data model, and can be applied to relational, Entity-Attribute-Value (EAV) or RDF data. In the $\tau - Space$, the *topological structure* of the data graph is preserved and represents the fine-grained component of the semantic model, which is grounded by the distributional semantics representation, which supports the semantic approximation operations using large-scale commonsense distributional knowledge.

The *distributional-relational representation* defines a *knowledge representation model for structured data* which supports the semantic matching between the query and the data. The representation model is complemented by a *query planning* approach which provides a *compositional model* for syntactically matching query to database structures. Figure 1.5 outlines the main components of the approach.

The first step (1) in the proposed approach is the construction of a *distributional semantic model* based on the extraction of word co-occurrence patterns from large corpora, which defines a distributional semantic vector space. The distributional semantic vector space uses concept vectors to semantically represent data and queries, by mapping dataset entities and query terms to vectors in the distributional space. Once the vector space is built, the structured data is embedded into the space (step 2), defining the $\tau - Space$, a distributional structured semantic vector space. The alignment between structured data and the distributional model allows the use of the *large-scale commonsense information* embedded in the distributional model (extracted from text) to be used in the *semantic matching/approximation* process.

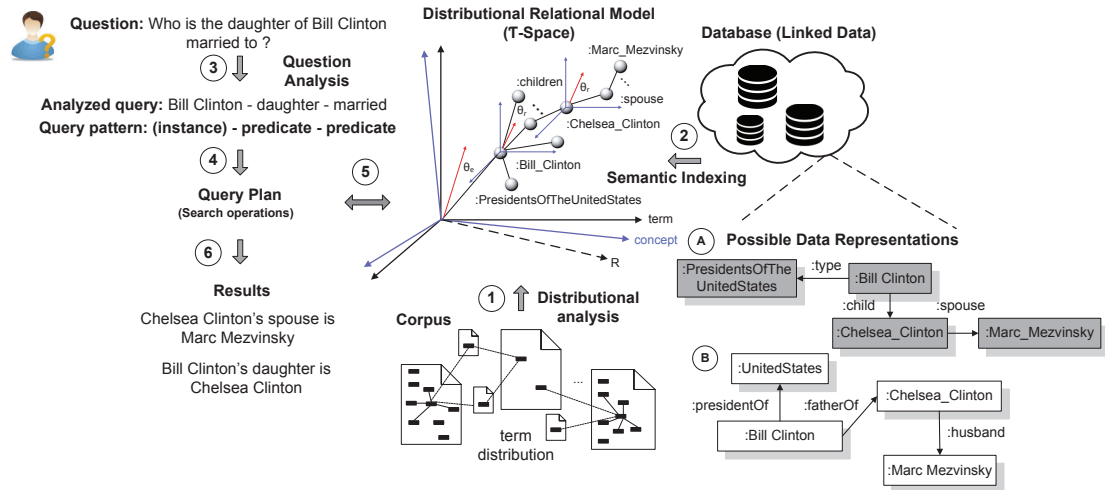


FIGURE 1.5: Schematic representation of the main components of the proposed query approach (for the example query in Figure 1.3). Numbers indicate the workflow from the construction of the distributional semantic model to the processing of the query results.

After the data is indexed into the τ – *Space*, it is ready to be queried. The query processing starts with the analysis of the natural language query, from which a set of *query features* and a *semi-structured query representation* is extracted (step 3). After the query is analyzed, a *query processing plan* is generated (step 4), which maps the set of features and the semi-structured query into a set of *search*, *navigation* and *transformation operations* (step 5) over the data graph embedded in the τ – *Space*. These operations define the semantic matching between the query and the data, using the distributional semantic information. This corresponds to the *compositional* model associated to the distributional model.

1.9 Open Domain vs. Domain Specific Semantic Matching

The demand for semantic matching approaches can be split into two main application scenarios with regard to the specificity of the data:

- *Open domain data:* Consists of data covering different domains of knowledge. Open domain data covers less specific information in different areas. The typical data consumer for the open domain data is the general Web user. Encyclopaedic scientific knowledge, commonsense, definitions, movies, sports, music, historical events, news are examples of domains typically covered in open domain datasets.

Examples of open domain datasets on the Web are DBpedia⁴, Freebase⁵, YAGO⁶ and CIAWorldFactBook⁷.

- *Domain specific data*: Data which describes a single domain, typically technical and with higher specificity. The typical data consumer for domain specific data is the domain expert or analyst. Financial reports, genomics and proteomics data are examples of domain specific data. Examples of domain-specific datasets are: PubChem⁸, Diseasesome⁹, Drugbank¹⁰.

The open domain scenario was selected to evaluate the proposed schema-agnostic query approach due to the: (i) more comprehensive evaluation of the semantic matching with regard to domain heterogeneity, (ii) larger schema size and (iii) better availability of evaluation campaigns and test collections.

More specifically, open domain Linked Open Data was selected to evaluate the proposed approach due to the availability of larger test collections and associated datasets, improving the comparability of the proposed model.

While the evaluation of domain-specific scenarios has a greater connection with real-world applications and, as a consequence, it presents larger application/utility potential, open-domain scenarios allows the empirical exploration of large-schema scenarios and more complex semantic phenomena (ambiguity, synonymy and vagueness). The lack of evaluation of the existing approach in domain-specific scenarios is a limitation of this work and should be taken into account while transporting the results of this thesis for domain-specific datasets.

1.10 Hypothesis

This thesis focuses on the corroboration of the following *core research hypothesis*:

- “*Distributional semantics* can be used to define a *low maintenance semantic model* which can support *schema-agnostic* queries over *open domain* structured databases.”

The *core research hypothesis* can be detailed into the following *research hypotheses*:

⁴<http://datahub.io/dataset/dbpedia>
⁵<http://datahub.io/dataset/freebase>
⁶<http://datahub.io/dataset/yago>
⁷<http://datahub.io/dataset/world-factbook-fu-berlin>
⁸<http://pubchem.ncbi.nlm.nih.gov/>
⁹<http://datahub.io/dataset/fu-berlin-diseasome>
¹⁰<http://datahub.io/dataset/fu-berlin-drugbank>

- **Research Hypothesis I:** Distributional semantics provides an accurate, comprehensive and low maintenance approach to cope with the abstraction and conceptual-level differences dimensions of semantic heterogeneity in schema-agnostic queries over large-schema open domain datasets.
- **Research Hypothesis II:** The compositional semantic model defined by the query planning mechanism supports expressive schema-agnostic queries over large-schema open domain datasets.
- **Research Hypothesis III:** The proposed distributional-relational structured vector space model ($\tau - Space$) supports the development of a schema-agnostic query mechanism with interactive query execution time, low index construction time and size, and it is scalable to large-schema open domain datasets.

The hypotheses can be directly mapped into the high coverage of the *core requirements for schema-agnostic queries* (Table 1.1).

Requirements	Hypotheses
High usability & Low query construction time	Hyp. I
High expressivity	Hyp. II
Accurate & comprehensive semantic matching	Hyp. I
Low setup & maintenance effort	Hyp. I
Interactive search & Low query-execution time	Hyp. III
High scalability	Hyp. III

TABLE 1.1: Mapping of the core requirements to the hypothesis.

1.11 Research Methodology

The research methodology in this thesis aims at providing a rigorous method of validating the hypotheses defined in the previous section. This thesis follows the research methodology described below:

1. Comprehensive literature survey of the state-of-the-art in the problem space.
2. Identification of a set of core requirements for a schema-agnostic query mechanism for databases with the support of the literature.
3. Categorization and gap analysis of existing works using the set of core requirements.
4. Definition of an evaluation dataset containing a categorized set of queries, reflecting the characteristics of open domain databases (large schema/schema-less with a large distribution of concepts).

5. Investigation of the semantic phenomena involved in the process of semantically mapping schema-agnostic queries.
6. Conceptualisation and formalisation of the schema-agnostic approach.
7. Implementation of the $\tau - Space$ semantic index and search.
8. Implementation of the schema-agnostic approach as a Natural Language Interface/Question Answering system over Linked Data scenario.
9. Design and execution of the evaluation.
 - Setup: Open domain question-answering system using the Question Answering over Linked Data (QALD) 2011 test collection over DBpedia.
 - Evaluation of the relevance of the results (measures: precision, recall, mean reciprocal rank);
 - Evaluation of performance & scalability (measures: query execution time and indexing time);
 - Evaluation of maintenance (measurements: time for dataset adaptation);
 - Comparative evaluation with existing approaches (previous measures compared against existing baselines);
10. Analysis of the results and conclusions.

1.12 Contributions

This work provides the following contributions:

1. Creation of a schema-agnostic query mechanism for large-schema open domain databases satisfying the core requirements:
 - Comprehensive and accurate semantic matching.
 - 80% of queries answered, avg. recall = 0.81, mean avg. precision = 0.62, mean reciprocal rank = 0.49.
 - Medium-high expressivity.
 - 80% of queries answered.

- Low maintenance.
 - 0 min adaptation effort.
 - 0.030/avg. disambiguation operations per query.
 - Interactive query execution time.
 - 8.530 s avg. query execution time.
 - Better recall and query coverage compared to baselines with equivalent precision;
 - 20% improvement of query expressivity, 21% of recall improvement compared to the existing best performing system while keeping equivalent precision (0.62).
2. Understanding of the changes in the database landscape which motivates schema-agnostic queries.
 3. Analysis of the existing literature with regard to schema-agnosticism.
 4. Improvement of the formal definition of schema-agnostic queries.
 5. Definition of a categorization framework for semantic complexity classes to resolve schema-agnostic queries (classes of *semantic resolvability*)
 6. Creation of a preliminary information theoretical model to evaluate the complexity of matching schema-agnostic queries.
 7. Creation of a distributional-relational vector space model ($\tau - Space$).
 8. Comprehensive evaluation of the proposed approach on the following dimensions, extending standard evaluation methodologies which are common practice in this field (QA/NLI):
 - Results relevance;
 - Performance (query execution time and index construction time);
 - Maintenance (time for dataset adaptation);
 - Comparative analysis with existing approaches;
 9. Implementation of a prototypical software infrastructure for the proposed approach.

- τ – *Space* semantic indexing and search engine
 - A high-performance Explicit Semantic Analysis (ESA) service
 - A Question-Answering (QA) System (Treo)
10. Generalization of the approach into a Knowledge-based Semantic Interpretation model (KBSI) and an associated Distributional Semantic Stack architecture.
 11. Application of the hybrid distributional-relational model into two scenarios: *selective reasoning over incomplete knowledge bases* and *logic programming*.

The core of this thesis concentrates on the proposal and evaluation of a schema-agnostic query model based on distributional semantic models. This core is complemented by a broader discussion which contextualises and generalises schema-agnosticism under contemporary data management environment.

1.13 Thesis Outline

The thesis is structured in the following chapters:

- *Chapter II - Semantic Heterogeneity & Schema-Agnostic Queries for Databases:* Analyses the changes in the database landscape, motivating how the growth in size, complexity, dynamicity and decentralisation of schemas (SCoDD) are bringing fundamental changes in data management, including the demand for schema-agnostic queries. Based on existing literature, the chapter also analyses the causes and dimensions of semantic heterogeneity between queries and databases. The dimensions which affect the resolution of schema-agnostic queries are categorized into a *semantic resolvability model*, which defines categories of semantic complexity for mapping schema-agnostic queries.
- *Chapter III - Literature Review:* Provides a description and analysis of the state-of-the-art for query mechanisms over heterogeneous databases using the set of core requirements for schema-agnostic query mechanisms. Different categories of query and search approaches are analysed including Natural Language Interfaces, Entity Search Engines and Approximate Query Mechanisms.
- *Chapter IV - Towards a New Semantic Model for Databases:* At the center of schema-agnostic query mechanisms is the definition of a semantic model which could cope with the semantic resolvability categories. The chapter provides an

analysis of the semiotic principles behind human-database communication and the associated semantic perspective on databases. Different perspectives on semantics (logical, cognitivist and structuralist) are analysed. Based on the analysis, a hybrid distributional-relational semantic model is outlined, targeting addressing the new semiotic assumptions which emerge in the *open communication scenario*.

- *Chapter V - The Semantic Matching Problem: An Information-Theoretical Approach:* Introduces a preliminary *quantitative information-theoretical model* for estimating the *semantic complexity of the query-dataset matching*. The goal of the entropy model is to provide a deeper understanding of the principles behind the schema-agnostic semantic matching problem, which is exploited on the design of the query mechanism, minimizing the semantic complexity of the matching process.
- *Chapter VI - τ -Space: A Hybrid Distributional-Relational Semantic Model:* Describes and formalizes the proposed hybrid distributional-relational ($\tau - Space$) semantic representation model supporting the schema-agnostic query mechanism. At the $\tau - Space$, each element in the data graph has an associated distributional semantic vector representation, which supports a geometric-based semantic approximation model, using the distributional knowledge. The structure of the $\tau - Space$ is defined by the mapping between data model categories and the associated distributional subspaces associated with each category.
- *Chapter VII - Distributional Semantic Search:* Describes the distributional semantic search approach in which the distributional semantic relatedness measure is used as a ranking function. The *semantic differential approach* for the determination of the threshold for the semantic relatedness-based ranking score is introduced, supporting the filtering of unrelated results.
- *Chapter VIII - The Schema-agnostic Query Processing Approach:* Describes the schema-agnostic query processing algorithm over the $\tau - Space$ distributional semantic model. The query processing approach users a set of semantic search, composition and data transformation operations over the $\tau - Space$. A supporting architecture for the query mechanism is proposed. The architecture is instantiated into the *Treo* prototype natural language query mechanism.
- *Chapter IX - Evaluation:* Describes the experimental methodology for the proposed schema-agnostic query mechanism, collects the evaluation metrics and analyses the results of the experiments.

- *Chapter X - Generalization & Further Applications:* This chapter explores developments and applications derived from the proposed τ – *Space* knowledge representation approach, with a particular focus on how it can be generalized to areas such as logical reasoning, in particular with regard to support selective and flexible reasoning over incomplete Knowledge Bases.
- *Chapter XI - Conclusion:* This chapter analyses the results on the evaluation of the research hypotheses, discusses the limitations of the schema-agnostic query approach and proposes a future work research agenda based on the limitations.

1.14 Associated Publications

Different aspects of this work were disseminated on the following publications:

- André Freitas, Edward Curry, Natural Language Queries over Heterogeneous Linked Data Graphs: A Distributional-Compositional Semantics Approach, In Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI), Haifa, 2014. (Full Conference Paper).
- André Freitas, João Carlos Pereira Da Silva, Edward Curry, Paul Buitelaar, A Distributional Semantics Approach for Selective Reasoning on Commonsense Graph Knowledge Bases, In Proceedings of the 19th International Conference on Applications of Natural Language to Information Systems (NLDB), Montpellier, 2014. (Full Conference Paper).
- André Freitas, Edward Curry, João Gabriel Oliveira, João C. Pereira da Silva, Sean O’Riain, Querying the Semantic Web using Semantic Relatedness: A Vocabulary Independent Approach. *Data Knowledge Engineering (DKE) Journal*, 2013. (Article).
- André Freitas, Edward Curry, João Gabriel Oliveira, Sean O’Riain, A Distributional Structured Semantic Space for Querying RDF Graph Data. *International Journal of Semantic Computing (IJSC)*, 2012. (Article).
- André Freitas, Edward Curry, João Gabriel Oliveira, Sean O’Riain, Querying Heterogeneous Datasets on the Linked Data Web: Challenges, Approaches and Trends. *IEEE Internet Computing*, vol. 16, no. 1, p.24-33, 2012. (Article).
- André Freitas, Edward Curry, Sean O’Riain, A Distributional Approach for Terminological Semantic Search on the Linked Data Web. In Proceedings of the 27th

ACM Symposium On Applied Computing (SAC), Semantic Web and Applications (SWA), 2012. (Conference Full Paper).

- André Freitas, João Gabriel Oliveira, Edward Curry, Sean O’Riain, A Multidimensional Semantic Space for Data Model Independent Queries over RDF Data. In Proceedings of the 5th International Conference on Semantic Computing (ICSC), 2011. (Conference Full Paper).
- André Freitas, João Gabriel Oliveira, Sean O’Riain, Edward Curry, João Carlos Pereira da Silva, Querying Linked Data using Semantic Relatedness: A Vocabulary Independent Approach. In Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems (NLDB), 2011. (Conference Full Paper).
- André Freitas, João C. P. da Silva, Sean O’Riain, Edward Curry, Distributional Relational Networks, AAAI Fall Symposium, Arlington, 2013 (Conference Paper).
- André Freitas, Rafael Vieira, Edward Curry, Danilo Carvalho, João Carlos Silva, On the Semantic Representation and Extraction of Complex Category Descriptors, In Proceedings of the 19th International Conference on Applications of Natural Language to Information Systems (NLDB), Montpellier, 2014. (Short Conference Paper).
- André Freitas, Edward Curry, Do it yourself (DIY) Jeopardy QA System, In Proceedings of the 12th International Semantic Web Conference (ISWC), Sydney, 2013 (Demonstration Paper in Proceedings).
- André Freitas, Fabricio de Faria, Sean O’Riain, Edward Curry, Answering Natural Language Queries over Linked Data Graphs: A Distributional Semantics Approach, In Proceedings of the 36th Annual ACM SIGIR Conference, Dublin, Ireland, 2013. (Demonstration Paper in Proceedings).
- André Freitas, Sean O’Riain and Edward Curry, A Distributional Semantic Search Infrastructure for Linked Dataspaces, In Proceedings of the 10th Extended Semantic Web Conference (ESWC), Montpellier, France, 2013. (Demonstration Paper in Proceedings).
- André Freitas, Sean O’Riain and Edward Curry, Crossing the Vocabulary Gap for Querying Complex and Heterogeneous Databases: A Distributional-Compositional Semantics Perspective, 3rd Workshop on Data Extraction and Object Search (DEOS), 29th British National Conference on Databases (BNCOD), Oxford, UK, 2013. (Abstract).

-
- André Freitas, João Gabriel Oliveira, Sean O’Riain, Edward Curry, João Carlos Pereira da Silva, Treo: Combining Entity-Search, Spreading Activation and Semantic Relatedness for Querying Linked Data, In 1st Workshop on Question Answering over Linked Data (QALD-1) Workshop at 8th Extended Semantic Web Conference (ESWC), 2011 (Workshop Full Paper).
 - André Freitas, João Carlos Pereira Da Silva, Edward Curry, On the Semantic Mapping of Schema-agnostic Queries: A Preliminary Study, Workshop of the Natural Language Interfaces for the Web of Data (NLIWoD), 13th International Semantic Web Conference (ISWC), Rival del Garda, 2014. (Workshop Short Paper)
 - André Freitas, Edward Curry, Siegfried Handschuh, Towards a Distributional Semantic Web Stack, 10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2014), 13th International Semantic Web Conference (ISWC), Rival del Garda, 2014. (Position Paper)
 - Danilo Carvalho, gatay lli, André Freitas, Edward Curry, EasyESA: A Low-effort Infrastructure for Explicit Semantic Analysis, In Proceedings of the 13th International Semantic Web Conference (ISWC), Rival del Garda, 2014. (Demonstration Paper in Proceedings)
 - André Freitas, João Gabriel Oliveira, Sean O’Riain, Edward Curry, João Carlos Pereira da Silva, Treo: Best-Effort Natural Language Queries over Linked Data, In Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems (NLDB), 2011 (Poster in Proceedings).

Chapter 2

Semantic Heterogeneity & Schema-Agnostic Queries

2.1 Introduction

The evolution of data environments towards the consumption of data from multiple *data sources* and the growth in the *schema size, complexity, dynamicity* and *decentralisation* (SCoDD) of schemas [54] increases the impact of *data heterogeneity* in contemporary data management. The SCoDD trend emerges as a central data management concern in Big Data scenarios, where users and applications have a demand for more complete data, produced by independent data sources, under different semantic assumptions and contexts of use [28]. While in the Big Data discourse *data variety* is used to describe the trend towards the availability and consumption of data from multiple data sources and from different types, the term *data heterogeneity* has a better grounding in the database literature, describing the *semantic* and *syntactic* dimensions in which data can vary [27].

The evolution of databases in the direction of heterogeneous data environments strongly impacts the *usability, semiotic* and *semantic assumptions* behind existing data accessibility methods such as structured queries. The main goal of this chapter is to provide a deeper understanding of the evolution of data management environments and a more rigorous and in-depth understanding of data heterogeneity and its relation to the semantic matching requirements behind schema-agnostic query mechanisms.

Section 2.2 concentrates on the analysis of the characteristics of modern data management environments and its impact on querying structured data. The growth in schema size and data heterogeneity fundamentally impacts the semiotic assumptions behind querying mechanisms, in which users traditionally need to interpret the schema and

manually reference database elements under a perfect symbolic and syntactic matching constraints. As this manual process becomes unfeasible for Big Data management environments, query mechanisms should evolve in the direction of automating the query-database semantic alignments, pointing in the direction of schema-agnostic query mechanisms. The understanding of the data heterogeneity dimensions supports the delineation of new requirements for structured query mechanisms.

In order to provide an analysis of semantic heterogeneity and semantic matching it is necessary to select a core *structured data model* for grounding the discussion. Section 2.4 provides a justification for the target data model aiming for maximizing the generality of the conclusions of this work. The selection of the reference data model takes into account: (i) its ability to represent or map other data models (generality/transportability), (ii) ability to map large and heterogeneous schemas, (iii) minimum number of syntactic constraints (simplicity). The use of a reference data model works as a necessary representation formality and does not limit the generality of the approach to other structured data models.

After selecting the reference data model, the phenomena of *data heterogeneity* is analysed. *Data heterogeneity* provides a framework to analyse the possible differences in datasets created under different contexts and requirements. Section 2.5 provides a description of the semantic heterogeneity dimensions. Based on the literature, the *causes of semantic heterogeneity* are investigated and a synthetical categorization of the *dimensions of semantic heterogeneity* is provided as a taxonomy.

Sections 2.7 and 2.8 describes the problems of semantic matching starting from the definition of *semantic tractability* introduced by Popescu et al. [55] in the context of natural language queries over databases. Since the concept of semantic tractability does not fully contemplate the spectrum of matching phenomena for the schema-agnostic query scenario, the concept of *semantic resolvability* and *semantic mapping types* are introduced.

2.2 Contemporary Data Management & Semantic Heterogeneity

2.2.1 Contemporary Data Management Environments

The database landscape is changing rapidly, influenced by the need to cope with databases with increasing schema size, complexity, dynamicity and decentralization (SCoDD) conditions. The emergence of new data sources such as open datasets on the Web, sensor

networks, data from mobile applications, social network data, together with the natural growth of datasets inside organizations [5], brings the demand for data management strategies which can operate under the properties dictated by this new data environment. Numerical indicators of this new data environment circa 2010 [5] include: 5 billion mobile users, 40% projected data growth in global data, 235 TB of data collected by the US Library of Congress in 2011, 15 out of 17 sectors in the US have more data stored per company than the US Library of Congress and 30 billion pieces of content per month shared on Facebook [1]. The challenges, new approaches and trends for coping with this new data landscape is currently aggregated under the *Big Data* umbrella term.

According to [56], Big Data can be defined as “*the term for a collection of datasets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications*”. Another definition for Big Data is given by the Gartner’s report [57] which provides a three-dimensional perspective of Big Data: “*Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization*”. More recently, the dimensions of *veracity* and *validity* were suggested to be added as characteristic features to the Big Data definition. Alternatively, Loukides [58] defines Big Data as “*when the size of the data itself becomes part of the problem and traditional techniques for working with data run out of steam*”. Along the same lines, Jacobs [59] states that Big Data is “*data whose size forces us to look beyond the tried-and-true methods that are prevalent at that time*”.

The value of Big Data can be described in the context of the dynamics of knowledge-based organisations [60], where the processes of *decision making* and *organizational action* (the *Knowing Cycle*) are dependent on the process of *sense making* and *knowledge creation*. At the basis of the sense making and knowledge creation processes is the *information seeking* behaviour, which is the process in which the individual purposefully searches for information that can change his or her state of knowledge or understanding [60], satisfying an information need. The increasing availability of data brings the opportunity of directly impacting the fundamental process of sense making and knowledge creation, allowing organisations and individuals to accelerate these processes. With increasing data at their hands, organizations and individuals have the necessary input to paint a more complete picture of their domain of interest, supporting better decision making and organisational action.

Information systems and users, already started to move in the direction of a more complete representation of different domains. In a 2010 survey, Brodie & Liu [5] report that database environments in Fortune 100 companies typically consist of tens of thousands of information systems with hundreds of databases per business area, where 90% of

them are relational, having a growth rate of 100s of databases per year. Typically, each database has between 100-200 tables, each table containing between 50-200 attributes. The number of views is typically three times the number of tables. In this data environment, the cost of integrating data across databases accounts for 40% of the software project costs. A comparative analysis shows the trend towards more complex and heterogeneous environments (Brodie & Liu [5]): while in 1985 a database would consist of two tables managed with a schema-based design, in 2010 there are 100-1000s of tables which are designed manually in an evidence gathering fashion, with 60-75% of these tables being *schema-less*.

2.2.2 The Growth of Data Variety

From the point of view of information systems and databases infrastructures, Big Data also represents the evolution in the direction of covering the *long tail of data variety* (Figure 2.1). The long tail of data variety reflects the distribution of the frequency of use of conceptual elements: in a large domain of interest few entities and attributes have a high frequency of use followed by a long tail distribution of entities and attributes which have lower frequencies of use. While some concepts are central across many different areas, most of the concepts are specific to a particular context. In the scientific domain for example, the *long tail of scientific data* [61] reflects the conceptual distribution of scientific data. From a historical perspective, the construction of information systems and databases has evolved following an *economic model* dependent on the *cost of formalizing a domain* and the *associated business value* derived from the efficiency gain. From an economic perspective, organisations have been prioritizing the formalization (conceptualization) of domains which accounted for recurrent transactions and/or demanded high governance.

Propelled by the growth of the Web and on the number of available computational devices, data management requirements are shifting towards the need to cope with *decentralised data generation* [7, 8]: data which is intrinsically heterogeneous, with different structuredness levels and generated under different meaning contexts. This scenario defines the availability of data under a long tail data variety distribution.

Relational databases provided a principled database model which allowed a precise and consistent representation, querying and manipulation of structured data. The first domains deployed in relational databases were typically transactional, were semantically homogeneous, having a relatively small number of databases, in which all views of the same entity were consistent, suiting naturally to the relational model [5]. For over

three decades relational databases have been the basis of information systems. Semantic homogeneity is key to the data modelling, querying and integration approaches that depend on the relational data model and ultimately its assumptions have constrained the modelling and deployment of databases. According to Brodie & Liu [5]:

“The consistency of all views of the same tuple leads the underlying belief in a single version of truth and the concept of a global schema. The dramatic success of relational technology has propelled data modelling and management requirements beyond the modelling and processing capabilities of the relational technology. The phrase ‘single version of truth’ seems intuitively correct and may provide assurance in a confusing world but is almost entirely false in the real world. The basic assumption of the relational world is not just semantic homogeneity but also ontological homogeneity while in reality semantic heterogeneity dominates. Data management vendors promote the ‘single version of truth’ assumption as a highly desirable objective and something that their products can provide. Our Digital Universe is no longer a semantically homogeneous set of a few databases but Information Ecosystems of 100s or 1,000s of semantically heterogeneous databases to be managed and integrated collectively.” [5].

To the growth of the SCoDD data conditions, it can be added the demand to integrate diverse interrelated data sources, with different data models, with distinct structural and conceptual granularities. To cope with diverse data coming from different sources it is necessary to deal with *semantic extensibility*, *semantic differences*, and *unknown semantics* [54]. In this scenario meaning is *variable*, *fuzzy* or *inconsistent*.

Franklin et al. [4] propose the concept of *dataspace* to describe this new type of data environment and its associated data management challenges. Franklin et al. [4] also points that *“in data management scenarios today it is rarely the case that all the data can be fit nicely into a conventional relational DBMS...”*. Under the *dataspace* scenario, the shift towards data co-existence instead of data integration is proposed.

The shift towards semantically heterogeneous environments is also emphasized at the Lowell Self-Assessment report [62], a roadmap for the future of research in the database community: *“A semantic heterogeneity solution capable of deployment at Web scale remains elusive... At Web scale, this is infeasible and query execution must move to a probabilistic world of evidence accumulation and away from exact answers. Therefore, one must perform information integration on the fly over perhaps millions of information sources”* [62].

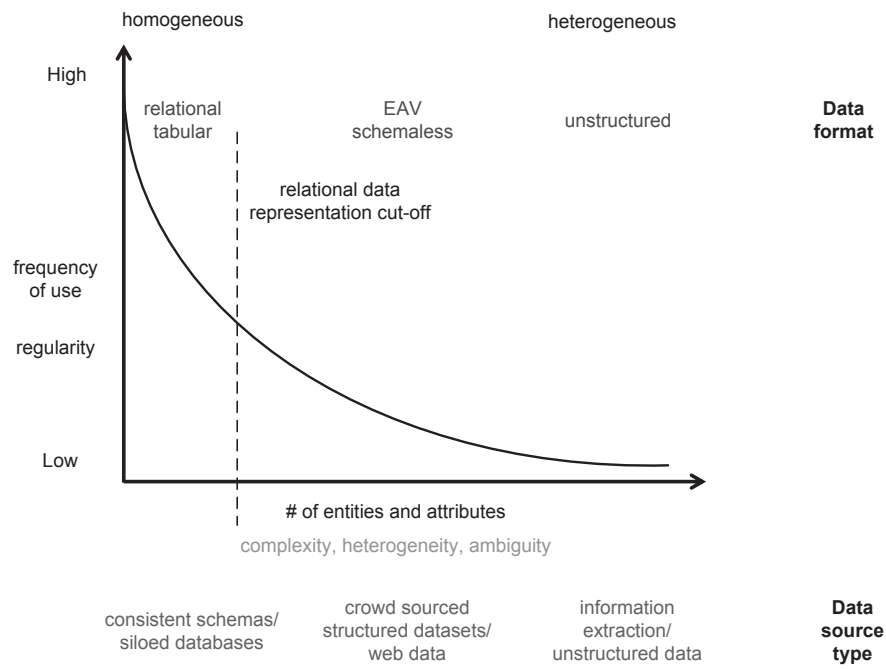


FIGURE 2.1: The long tail of data variety.

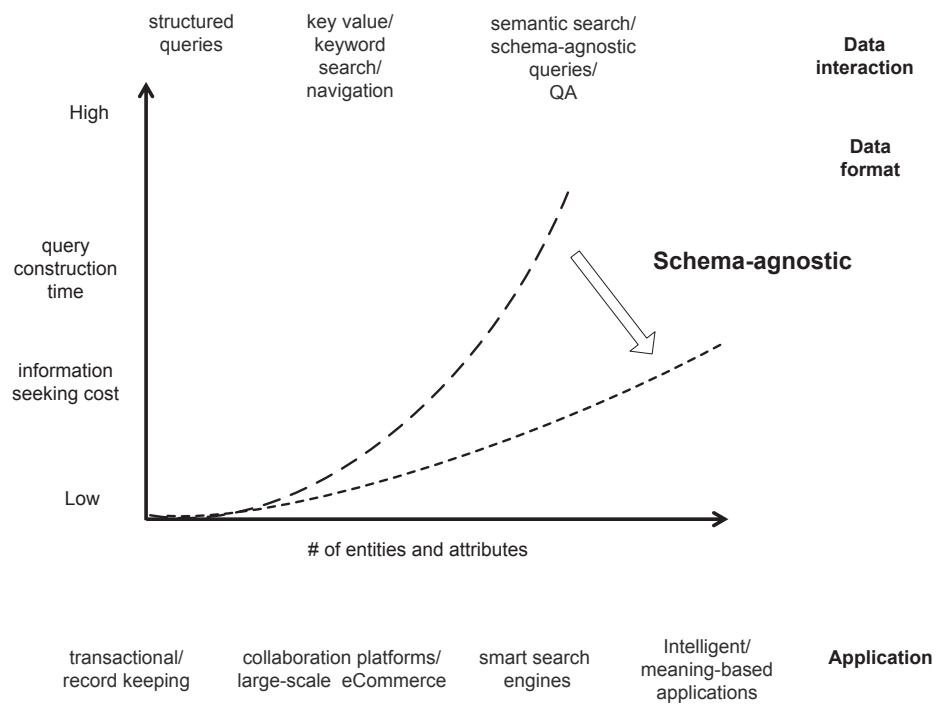


FIGURE 2.2: Data consumption cost associated with schema size.

2.2.3 Schema-less Databases

The pressure towards the provision of principled solutions for the challenge of managing semantically heterogeneous data is also present from the data modelling perspective. It is recognised [63, 64] that the problem of data modelling encompasses issues that may not be amenable to formalization, and that the supporting practices are not followed in reality [63]. This is corroborated by Brodie & Liu [5] which reported in their survey that while 90% of all information systems inside 10 Fortune 100 companies are relational, no single instance of an entity-relationship model was found. Modelling tools also may be unable to cope with fast-varying world states [65].

In this context Badia argues that, “*in a very real way, we have entered a post-methodological era as far as the design of information systems is concerned. The emergence of the Web has coincided with the death of the dominant methods based on the analytic thought and lead to the emergence of sense-making as a primary paradigm*”. This points to the centrality of *search* as an evidence gathering activity in the Web scenario. These facts show that data environments are evolving from rigidly defined *prescriptive schemas* (where the data is forced into a fixed semantic model, in the form of a semantic contract) to a *descriptive model* (where the author describes what is intended in the schema) [54], which provides more adaptable, flexible, and extensible approaches [63]. As Helland notes [54], increasingly, schema definition is captured in the ‘name’ of a name/value pair in a descriptive fashion: we are moving from SQL DDL to XML/pair-value. As systems evolve in the direction of large-scale, multi-domain and distributed systems, *adaptability* and *flexibility* offer more value than *crispness* and *clarity* [63].

The ability to naturally evolve and extend the model and also to support information incompleteness is an important requirement in this new scenario. This requirement points into the direction of a data model perspective which embraces an *open world assumption* [63]. With this level of flexibility and extensibility, models could be created collaboratively [63], supporting database designers to address the long tail of data variety. This need of extensibility and dynamic schemas is supported by the emergence of *schema-less platforms*.

2.3 The Vocabulary Problem & Schema-agnostic Queries

2.3.1 The Vocabulary Problem for Databases

The vocabulary problem [9] is a recurrent problem in the communication between humans and information systems, where humans’ *information needs* and *intents* need to be

mapped to symbols accessible to users through the system interface. The symbols refer to different computational resources and structures, such as database schema elements, commands, configuration parameters, filenames, among others. According to Furnas et al. [9]:

“... Many functions of most large systems depend on users typing in the right words. New or intermittent users often use the wrong words and fail to get the actions or information they want. This is the vocabulary problem. It is a troublesome impediment in computer interactions both simple (file access and command entry) and complex (database query and natural language dialog).”

Furnas et al. analyses the vocabulary usage variability in 6 tasks, where different populations of users are asked to name objects and actions in different domains. The domains involved information objects that a generic user might want to access on a computer and were all of modest size (5-200 objects) [9]. Furnas et al. found that if a person assigns the name of an information resource, other untutored people will fail to access it on 80% to 90% of their attempts [9]. Their experiments focused on scenarios where the set of access terms items are distinct. In many scenarios, there is a high likelihood that there is an overlap between different keywords present in multiple access terms, introducing ambiguity and decreasing the probability of a correct term selection. Under the same tasks, improved naming schemes based on popularity account for failures in 65-85% of the time. Furnas et al. propose unlimited aliasing as the solution for untutored vocabulary driven access where many different aliases for access terms are provided. They conclude that *“... Thus aliases are, indeed, the answer, but only if used on a much larger scale than usually considered.”* [9].

Furnas et al. [9], however, explored scenarios which are limited in the number of objects and on their compositional/descriptive complexity (number of words to describe a resource). The impact of the vocabulary problem in a specific human-system communication scenario is dependent on the complexity of the system vocabulary (number of symbols, possible combinations between symbols). The increasing availability of computational resources, the emergence of the Web and of mobile platforms, the increasing accumulation of software artefacts and data, brings the vocabulary problem today to a new scale.

The process of schema-agnostic database querying for large-schema databases is a challenging instance of the vocabulary problem.

2.3.2 Schema-agnostic Queries

Schema-agnostic queries can be defined as query approaches over structured databases which allow users satisfying complex information needs without the understanding of the representation (schema) of the database. Similarly, Tran et al. [6] defines it as “*search approaches, which do not require users to know the schema underlying the data*”. Approaches such as keyword-based search over databases allow users to query databases without employing structured queries. However, as discussed by Tran et al [6]: “*From these points, users however have to do further navigation and exploration to address complex information needs. Unlike keyword search used on the Web, which focuses on simple needs, the keyword search elaborated here is used to obtain more complex results. Instead of a single set of resources, the goal is to compute complex sets of resources and their relations.*”

Despite being present as an implicit requirement for different types of query mechanisms, schema-agnostic queries and their semantic matching implications have not been explicitly defined in the existing literature. This work fills this gap, by analysing the semantic heterogeneity dimensions and the semantic matching problem under the schema-agnostic query scenario.

The development of approaches to support natural language interfaces (NLI) over databases have aimed towards the goal of schema-agnostic queries. Complementarity some approaches based on keyword search have targeted keyword-based queries which express more complex information needs. Other approaches have explored the construction of structured queries over databases where schema elements can be relaxed. All these approaches (natural language, keyword-based search and approximate structured queries) have targeted different degrees of sophistication in addressing the problem of supporting a flexible *semantic matching* between queries and data, which vary from the completely absence of the semantic concern to more principled semantic models, the latter usually developed in the scope of the Natural Language Processing (NLP) community.

Most discussion on semantic matching for schema-agnostic queries has targeted a systems perspective (as in the NLI/QA over databases literature). This work argues that schema-agnostic queries demand new perspectives of semantics for databases, including new *semantic models* and associated *semantic matching algorithms*. The proposal of a new semantic model to support schema-agnostic queries is the object of investigation of Chapters 4 and 6.

2.3.3 Querying Semantically Heterogeneous Data

This new database perspective which is shaped by the demands of coping with contemporary data management challenges, should be seen as a complementary perspective to the classical database perspective, responding to complementary demands.

Classical relational databases offer crisp and accurate answers for relatively small amounts of homogeneous data [54], typically managed in a centralised way. This paradigm defined the use of structured query languages (such as SQL) as the primary way to interact with the data. Below, the assumptions related to the user-data interaction behind structured queries are revisited:

1. *Clean, semantically homogeneous & centralized schema:* As schemas are managed in a decentralised way, different conceptualisations may exist in the same schema. “*We can no longer pretend to live in a clean world*” [54]. “*Unless the reader of a message or document is specifically programmed for it, there will likely be confusion. The meaning of the message, the interpretation of its fields, and much more will be subject to approximation and a loss of clarity. Different companies, different countries, and even different regions within a country have different understandings of data*” [54].
2. *Manual query-schema mapping:* Most of the interaction with structured data is dependent on a manual mapping between elements of a structured query and schema elements. With the growth in the schema-size and in the number of available data sources, the cost associated with this manual mapping process becomes prohibitive (Figure 2.2).
3. *Absolute precision/full recall in a single query:* As schemas grow and as users cross databases boundaries, the cost associated with building structured queries exponentially grows. In this scenario the expectation of getting a correct and complete answer in a single interaction should be exchanged by approximate answers which are obtained through multiple interactions. As Helland states [54] “*Too much, too fast-you need to approximate*”.
4. *Rigid data access view:* Typically users access databases with the help of a structured query language such as SQL or SPARQL or mediated by domain-specific applications which provide the interface to pre-defined structured queries. The mediation through a domain specific application provides a constrained view over structured data under a specific context of use. Providing direct query capabilities for users, maximizes the utility of data under different contexts of use.

The change of these assumptions deeply impacts the usability, semiotics and semantics perspectives of databases. From a usability perspective, users should be able to interact with databases under a schema-agnostic perspective, i.e., being abstracted from the representation of the data. A schema-agnostic query mechanism depends on a new perspective of semantics for databases which depends on the analysis of three dimensions:

1. Revisit the semantic homogeneity assumptions behind databases, providing a model which is able to capture and represent semantic heterogeneity.
2. Provide a model which can support a universal schema-agnostic query-database semantic matching mechanism.
3. Analyze the interaction aspects implied by this new semantic model, i.e. shifting from the expectations of absolute answer completeness and correctness in a single query to an approximate multi-stage interaction approach.

These dimensions are expressed in [5], [54], [4] and are better individuated in this work in the context of schema-agnostic queries. The construction of this supporting model is analysed in Chapter 4.

2.4 Data Models

2.4.1 Introduction

Different databases abstractions have been created to cope with different representational and operational demands. The multiplicity of data model categories represents a natural specialisation to cope with different data management requirements. Under the scope of this thesis, the analysis, the formulation and evaluation of a schema-agnostic query approach is grounded on a *reference data model*. In order to maximize the utility of the associated scientific results, the selection of the reference data model obeys the following criteria:

1. **Ability to represent large, complex & heterogeneous conceptual models:** Associated with the representation trends in data management and with the data environment which motivates schema-agnostic queries.
2. **Ability to represent and map other data models:** Maximizing the generality and transportability of the results.

3. **Adoption in test collections suitable for schema-agnostic queries:** Maximizing the comparability of the query approach.

Below the selection of the reference data model is described and analysed according to the criteria defined above.

2.4.2 Relational Databases

The *relational model* for database management is a data model based on first-order predicate logic, formulated and proposed by Codd [66]. In the relational data model data is represented as tuples and grouped into relations. The motivation behind the relational model is to provide a declarative method for specifying data and queries. The relational model emerged to define a model for describing data in terms of its natural structure, without superimposing any additional data management structure. According to Codd [66], “*it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and representation and organization of data on the other*”.

The relational model’s central idea is to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values. The content of the database is a finite logical model of the database, i.e. a set of relations, one per predicate variable, such that all predicates are satisfied.

The fundamental assumption of the relational model is that all data is represented as mathematical n -ary relations, an n -ary relation being a subset of the Cartesian product of n domains. In the mathematical model, reasoning about the data is done in two-valued predicate logic, meaning there are two possible evaluations for each proposition: either true or false. Data are operated upon by means of a relational algebra.

In the relational model a *tuple* is an ordered set of *attribute values*. An *attribute* is an ordered pair of *attribute name* and *type name*. An attribute value is a specific valid value for the type of the attribute which can be either a scalar value or a more complex type [66].

A *table* is a visual representation of a *relation* and a *row* maps to the concept of a *tuple*. The consistency of a relational database is enforced, not by rules built into the applications that use it, but rather by constraints, declared as part of the logical schema and enforced by the Database Management System (DBMS) for all applications [66].

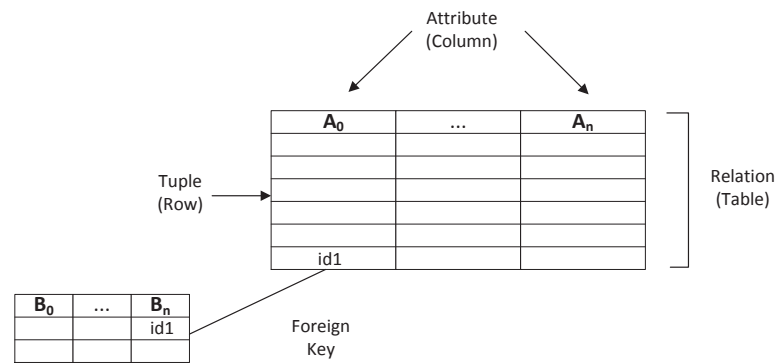


FIGURE 2.3: Visual representation of elements of the relational model.

Despite its major adoption, relational databases have limitations for coping with the list of data model requirements. These limitations are:

- **Ability to represent large, complex & heterogeneous conceptual models:**

- *Limitation in the representation of sparse data.* Relational models target the representation of compact (non-sparse) data. A relation is defined by a rigid set of attributes where the state of each attribute in a relation needs to be defined by a value assignment, for each tuple. The representation of relations which have larger set of attributes with optional value assignments demands the DBMS to manage these optional attribute assignments. Additionally, the table visual abstraction is not appropriate for relations containing potentially thousands or millions of attributes.
- *Schema rigidity.* Relational models are based on the concept of a prescriptive schema, which enforces the consistency of the data under a specific conceptual model. Relational models constrain the evolution of the schema.
- *Complex data integration:* The integration of relational databases under different schemas depends on a redefinition of the schema.

- **Ability to represent and map other data models:** Mapping is limited by the maximum number of attributes for a table.

- **Adoption in test collections suitable for schema-agnostic queries:** Tang & Mooney [67] provided the main test collection for natural language over databases, covering three domains. Its main limitation is schema size (for example the geography database contains 9 relations, 28 attributes).

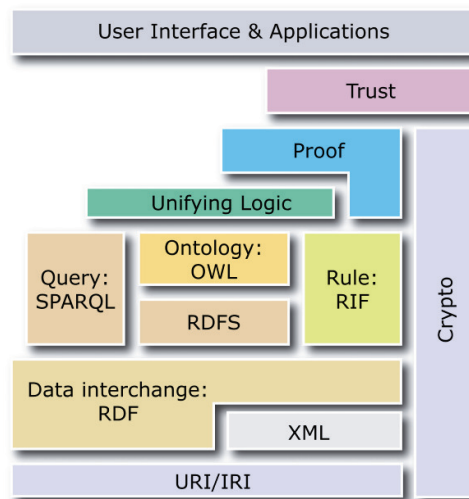


FIGURE 2.4: Layered Semantic Web model.

2.4.3 Semantic Web Databases & Linked Data

New data models emerged to facilitate the deployment of decentralized, large-schema, sparse and schema-less data environments. More prominently the effort associated with the creation of a Semantic Web / Linked Data Web have motivated and catalyzed the development of standardized data models which can support the SCoDD conditions.

The Semantic Web is based on the vision of the construction of a Web-scale distributed data representation and reasoning infrastructure which could support the development of intelligent applications and agents [68]. The vision of the Semantic Web is grounded on a standards-based data representation, consisting of layers of different knowledge representation concerns (Figure 2.4 [69]¹). The bottom layers cover the definition of identifiers (URIs), a standardized graph data model (the Resource Description Framework (RDF)) and its associated serialization format under the eXtensible Markup Language (XML), its typing and taxonomical system (RDF-Schema (RDF(S)) and the associated structured query language (SPARQL) [29]. The upper layers cover the representation of logical constructs (Web Ontology Language (OWL) [70]) and rules representation (Rules Interchange Format (RIF) [71] and the Semantic Web Rule Language (SWRL) [72]).

The problems associated with ensuring logical consistency and reasoning performance at Web-scale, strongly limited the adoption and growth of the Semantic Web. In order to increase its adoption, Berners-Lee [73] proposed a simplification over the previous Semantic Web representation scheme, by concentrating on the data model representation layers (lower layers of the Semantic Web stack), focusing on the mapping and publication

¹Image taken from <http://www.iro.umontreal.ca/~lapalme/ForestInsteadOfTheTrees/HTML/ch07.html>

of existing datasets available in data silos. This simplification defined the vision of a Linked Data Web, targeting the creation of a Web of structured data based on a standardized data model (RDF(S)). Since the conception of the Linked Data Web vision, datasets covering different domains have been exposed as Linked Data (Figure 2.5), showing a consistent adoption curve [74].

Linked Data provides a framework for a decentralised pay-as-you-go structured data integration, where the minimum level of integration is provided by the standardised data model representation and where the URIs and the Domain Name Systems (DNS) provide a global-level identification scheme, which facilitate the referencing of data entities among different datasets. Differently from the relational model, which depends on a *schema-level data integration*, the Linked Data Web is based on an *entity-centric data integration model*, where URIs representing objects or concepts can be reused or reconciled among different datasets. The entity-centric data integration facilitates the co-existence of different perspectives and points of views of entities and a decentralized evolution of the data. Complementarily, the use of *Linked Data vocabularies*, the specification of conceptual models for a domain under the RDF(S) model, are used to facilitate the interoperability and semantic integration among different datasets for specific domains. Vocabularies are used under a *descriptive*, instead of a *prescriptive* perspective: instead of being a semantic contract as in the relational model, the vocabulary works as a semantic recommendation for a conceptualization.

The publication of Linked Data is summarized under the Linked Data principles [75]:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look-up those names.
3. When someone looks-up a URI, provide useful information, using the standards (RDF(S), SPARQL)
4. Include links to other URIs, so that they can discover more things.

Linked Data is an entity-centric model which allows an entity-based data integration process which, together with the Web standards of RDF(S), URI and HTTP, defines a de-facto integration model for the Linked Data Web.

The adoption and publication of Linked Open Data on the Web by different organizations have created a set of concrete instances of SCoDD data environments (Figure 2.5) which are publicly available. These datasets provide a clear picture of the problems which emerge under the SCoDD properties and opens a public space for research and

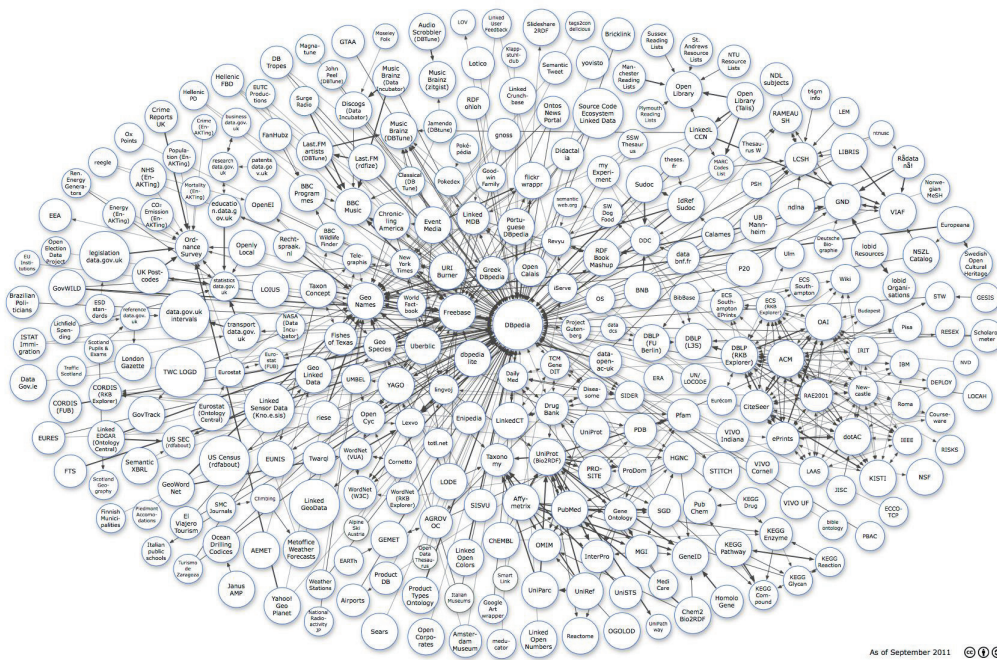


FIGURE 2.5: Datasets available in the Linked Data Cloud circa 2011.

experimentation. For this reason, Linked Data is being adopted in the evaluation of contemporary data environment challenges such as, *question answering*, *entity search*, *entity consolidation*, among others.

- **Ability to represent large, complex & heterogeneous conceptual models ([74, 75]):**
 - *Support for the representation of sparse data:* RDF(S) is based on a graph data model, which supports a sparse data model.
 - *Schema flexibility:* RDF(S) datasets are schema-less and can be evolved in a decentralised manner.
 - *Entity-centric integration:* Support for decentralised entity-centric data integration. From the perspective of small data publishers, the publication of data under the Linked Data principles represents an overhead with regard to the use of existing tools and formats (e.g. spreadsheets and csv files). In this scenario, Linked Data and RDF(S) can be seen as a data integration layer where there is a mapping between tabular files and RDF(S).
- **Ability to represent and map other data models:** Data in a relational or in a CSV format can be systematically mapped to RDF [76].

- **Adoption in test collections suitable for schema-agnostic queries:** Various test collections for Linked Data are available, under SCoDD conditions based on datasets with 1,000s or 1,000,000s of attributes: Question Answering over Linked Data [77], Semantic Search Challenge [78, 79], INEX Question Answering track [80].

The generality of RDF(S) in the representation of data under the SCoDD characteristics, the ability of RDF(S) to map existing data models, including relational databases, csv files, datalog, key-value pairs, the availability of real world datasets, the possibility of describing complex conceptual models using RDF Schema, and its support as a data model for a logical model, motivated the use of RDF(S) as the core data model for grounding this work.

This thesis focuses on the use of the minimum number of RDF(S) data model constructs, which are described in the next section.

2.4.4 The RDF(S) Data Model

Definition (RDF Triple). Let U be a finite set of URI resources, B a set of blank nodes and a L a finite set of literals. A triple $t = (s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$ is an RDF triple where s is called the *subject*, p is called the *predicate* and o the *object*.

Definition (RDF Graph). An RDF graph G is a subset of G , where $G = (U \cup B) \times U \times (U \cup B \cup L)$.

RDF Schema (RDF(S)) is a semantic extension of RDF. By associating predicates to RDF elements such as *rdfs:subClassOf*, *rdfs:subPropertyOf*, *rdfs:domain*, *rdfs:range*, *rdfs:Class*, *rdfs:Resource*, *rdfs:Literal*, *rdfs:Datatype*, etc., RDF(S) allows to express simple taxonomies and hierarchies among properties and resources, as well as domain and range restrictions for properties. The following definitions based on the notation of Eiter et al. [81] provides summarized description of RDF(S). A more complete formalization of the RDF(S) Semantics can be found in [81].

Definition (Class). The set of classes C is a subset of the set of URIs U such that $\forall c \in C$:

$$\forall c(\text{triple}(c, \text{rdf} : \text{type}, \text{rdfs} : \text{Class})) \supset \text{triple}(c, \text{rdfs} : \text{subClassOf}, \text{rdfs} : \text{Resource}) \quad (2.1)$$

Definition (Domain and Range). The *rdfs:domain* and *rdfs:range* of a property *p* in the triple *t* in relation to a class *c* are given by the following axioms:

$$\forall s, p, o, c(\text{triple}(s, p, o)) \wedge \text{triple}(p, \text{rdfs} : \text{domain}, c) \supset \text{triple}(s, \text{rdf} : \text{type}, c) \quad (2.2)$$

$$\forall s, p, o, c(\text{triple}(s, p, o)) \wedge \text{triple}(p, \text{rdfs} : \text{range}, c) \supset \text{triple}(o, \text{rdf} : \text{type}, c) \quad (2.3)$$

Definition (Instances). The set of instances *I* is a subset of the set of URIs *U* such that $\forall i \in I$:

$$\forall i(\text{triple}(i, \text{rdf} : \text{type}, \text{rdfs} : \text{Class})) \supset \text{triple}(i, \text{rdf} : \text{type}, \text{rdfs} : \text{Resource}) \quad (2.4)$$

2.4.5 Entity-Attribute-Value(EAV/Classes & Relations (CR))

RDF(S) can be abstracted into an *Entity-Attribute-Value* (EAV) data model.

“The *EntityAttributeValue* model (EAV) is a data model to describe entities where the number of attributes (properties, parameters) that can be used to describe them is potentially vast, but the number that will actually apply to a given entity is relatively modest” [82]. The EAV model can be defined as a sparse matrix where only non-empty values are stored [82]. From a practical perspective the EAV model is associated with open/-dynamic schema databases. EAV can be seen as the more abstract data model behind RDF(S) and Linked Data. An EAV data model is composed of three core elements [82, 83]:

- **entity:** the element being described. In RDF(S) the *entity* maps to an *instance*.
- **attribute:** the attribute definition. In RDF(S) it an *attribute* maps to a *property* or a *class*.
- **value:** The value assigned to an attribute. In RDF(S) it maps to an object which can be an *instance* or *value*.

On the top of the EAV abstraction, RDF(S) also defines a canonical ordering of a triple (*s*, *p*, *o*) and a *rdfs:type* relation which is used to assigning a unary predicate (class) to

RDF(S)	EAV/CR	Relational	Logical
Instance	Entity	Value	Constant
Value	Value	Value	Constant
Class	Attribute	Relation	Unary predicate
Property	Attribute	Attribute	Binary predicate

TABLE 2.1: Correspondence between the categories of the RDF(S), EAV, Relational and First-order logic data models.

an instance. An EAV model with the characteristics above is named a EAV/CR (EAV with Classes and Relationships).

2.4.6 Data Model Mappings

Table 2.1 describes the mappings between RDF(S), EAV, Relational data models. The corresponding first-order logical elements are included as an abstract representation but not discussed as a data model. This mapping provides a high-level correspondence between the elements in the data model, informally defining a transportability between the different data models.

2.5 Semantic Heterogeneity

2.5.1 Introduction

A *database model* is a simplified representation of objects from the real world. The database model materializes a *human conceptualisation* (conceptual model) which is defined by a specific set of system requirements, under a database representation framework (Kashyap & Sheth, 1996 [27] and Garcia-Solaco et al, 1993 [84]). The conceptual model reflects a subset of this conceptualisation, representing *individuals*, their *categorisations*, *properties* and *relationships*.

Semantic heterogeneity in databases represent differences in the real world interpretation of context, meaning, and use of data and occur during the designers task of translating conceptualisations of the real world into database-level representations. It reflects data model, schema construction, and data inconsistencies in the conceptual and database worlds (Kim et al., 1993 [85], Hammer and McLeod, 1993 [86], Kashyap & Sheth, 1996 [27], Garcia-Solaco et al., 1996 [84]).

2.5.2 Intrinsic Causes of Semantic Heterogeneity

This section provides a synthesis of the conceptual framework for modelling the causes behind semantic heterogeneity in databases. The works of (George, 2005 and Sheth & Larson, 1990 [87, 88]) are the main references on the investigation of the intrinsic causes of semantic heterogeneity. The list below summarizes the main causes for semantic heterogeneity according to [87, 88]:

1. *Design Autonomy*: Design autonomy can be reflected in differing designer influences and perception of the universe of discourse, data model representation (model and query language), naming conventions, semantic interpretation of data, and constraints applied (Batini et al., 1986 [89], Sheth & Larson, 1990 [88]).
2. *Development Autonomy*: “*Islands of development occur where organisations have evolved as collections of distinct, autonomous departments with disconnected systems; each pursuing its own IT infrastructure*” (Lamb & Davidson, 2000 [90]). Alternatively, a database structure may be simply too complex to be modelled by one designer.
3. *Different Universes of Discourse (Context)*: The interpretation of a term is intrinsically dependent on the universe of discourse (context) where it occurs. In databases and information systems, the context may not be explicitly and formally represented within the scope of a system (it can be defined in other systems or artefacts or at the human user level).

In addition to the intrinsic causes, there are *environmental impact factors* which increase the probability of semantic heterogeneity:

1. *Schema size*: The number of concepts expressed in the conceptual model and materialized in the database model.
2. *Conceptual complexity*: The number of different domains and concepts expressed in the conceptual model.
3. *Domain variability*: Degree of subjectivity in the description of a domain. Number of possible representations in the description of the conceptual model.
4. *Conceptual dynamics in the domain*: The amount of changes in the conceptual model over a period of time.

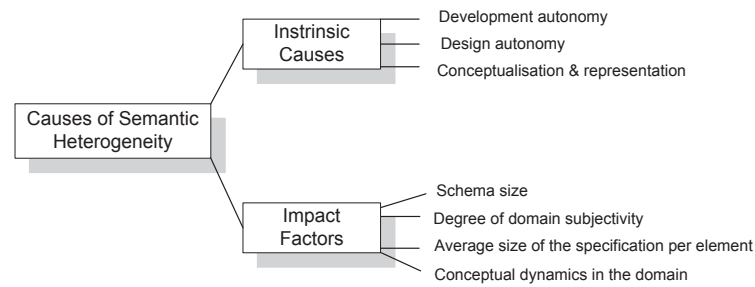


FIGURE 2.6: Intrinsic causes of semantic heterogeneity.

The presence of the environmental factors provide the indication of a higher probability of semantic differences between datasets or between query and data. Figure 2.6 depicts the causes of semantic heterogeneity.

2.6 Dimensions of Query-Database Semantic Heterogeneity

Most of the analysis on semantic heterogeneity have been done in the context of data/schema integration providing a comprehensive analysis of the dimensions involved in semantic heterogeneity between two datasets. Different works modelled data semantic heterogeneity (Colomb, 1997 [91], Parent & Spaccapietra, 1998 [92]). Other works defined classifications focusing on schema conflicts (Sheth & Kashyap [27]). Sheth & Kashyap [27] and George [87] provide comprehensive semantic heterogeneity taxonomies which grounds the semantic heterogeneity discussion of this work. The problem of semantically matching a schema-agnostic query and a database has commonalities to the problem of aligning elements between two datasets. The specificity of query-database alignments, however, lies on the asymmetry between the level of available contextual information and different structural levels between query and database.

The classification of the semantic heterogeneity or query-database semantic differences is fundamental for understanding the challenges that a semantic mechanism supporting a schema-agnostic query should cope with. This section discusses and classifies the dimensions of semantic heterogeneity in the context of the gap between query and database, organizing them into a taxonomy of query-database *semantic differences*. The construction of the taxonomy of query-database differences is guided by the following methodology:

1. Listing of concepts expressed in the existing semantic heterogeneity taxonomies (George [87], Colomb [91], Parent & Spaccapietra [92], Sheth & Kashyap [27]).
2. Elimination of concepts which were not relevant in the context of the query-database semantic differences.
3. Alignment between concepts which are equivalent.
4. Merging and renaming of equivalent concepts.

The categories for the taxonomy for query-database lexico-semantic differences are described below. The taxonomy categorizes different types of semantic differences between query terms and corresponding database elements, assuming that there is a valid semantic mapping between them. Figure 2.7 shows the taxonomy of query-database semantic differences, while Figure 2.8 shows different examples query-database semantic differences.

1. *Synonym*: Different lexical expressions mapping to the same concept (e.g. *customer* → *client*).
2. *Lexical Differences*: Lexical expressions with the same stem or with similar strings mapping to strongly related concepts.
3. *Conceptual Differences*: Distinct but related concepts under different lexical expressions in which the alignment satisfies the query information need.
 - (a) *Taxonomical Differences*: Differences in the core abstract structures between the database representation and the abstraction used in the query. ‘*PresidentOfTheUnitedStates*’ and ‘*AmericanPoliticians*’ express two different sets where the former set is contained in the second. In some cases the abstraction level expressed in the query may be different from the dataset and only a semantically approximate result can be returned. In this case users may need to verify the suitability of the approximation. Two entities are *semantically similar* if they are under the same taxonomical structure.
 - (b) *Non-taxonomical Differences*: A concept in the query and a concept in the database can represent distinct but strongly related concepts in the context of the query. For example the correspondence between ‘*married*’ and ‘*spouse*’. Two entities are *semantically related* if they have a non-taxonomical and non-synonymic semantic relationship.

4. *Compositional/Predication Differences*: Information may be expressed as different compositions of different database elements or predicate structures. ‘*PresidentsOfTheUnitedStates*’ can be expressed as a single predicate or as a composition of the binary predicate ‘*president*’ and the instance ‘*UnitedStates*’.
5. *Functional Differences*: Aggregated information may be already conceptualised in the database or may need to be computed based on existing data. For the example query in Figure 2.8(3), the predicate ‘*numberOfKids*’ could be expressed directly on the database or may need to be computed as an aggregation function over statements containing the predicate ‘*child*’. Superlatives are also examples of concepts which can be expressed either as predicates or through functions (e.g. ‘*highest*’ mapping to ‘*elevation*’) in Figure 2.8(6).
6. *Convention Differences*: Consists of differences in the conceptualisation of the values and units used (RGB vs. HSV color scheme), numerical vs. non-numerical, dates, dimension, units of measure and scale differences (units of measure, volume, weight, size, currency, e.g. miles vs. kilometres, Celsius vs. Fahrenheit), unique identifiers (employer name + birthdate vs. employer ID).
7. *Null Mappings*: Consists of a null mapping from a query term to a database element or vice-versa.
8. *Intensional Differences*: Consists of different intensional definitions expressed by the same term. An intensional definition consists of the properties that a term must satisfy. The definitions for ‘*taxable revenue*’, ‘*age of majority*’ and ‘*economically active population*’ are concepts which are likely to vary between different regions, and although representing similar concepts they have different definitional properties (for example ‘*economically active population*’ might be defined by different combinations and values associated with *age limit*, *job status* and *minimum salary*).
9. *Contextual Differences*: Consists of differences in the context in which an alignment holds. The predicate ‘*most awarded actor*’ can vary for different time spans and countries.

The classification previously described focuses on a single language and single data model query scenario. Schema-agnostic queries might include cross-language and cross-data models queries.

In order to address the vocabulary problem, schema-agnostic query approaches depend on the ability to match queries to database elements. The next sections formalizes the

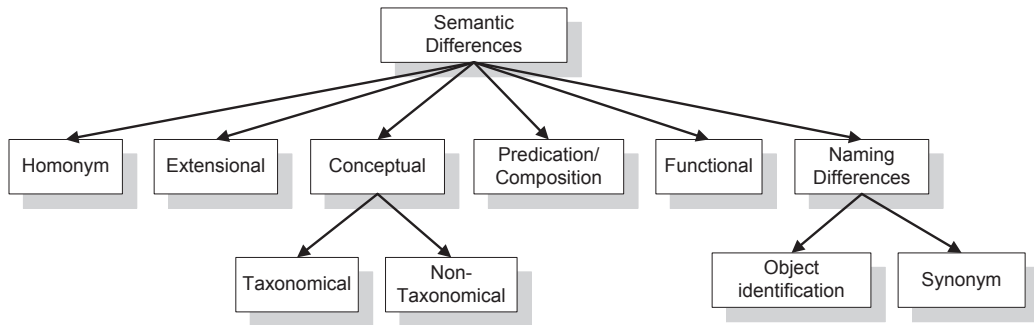


FIGURE 2.7: Taxonomy of lexico-semantic differences.

problem of semantic matching using the concept of *semantic tractability* developed by Popescu et al. [55] and its extension proposed in this work (*semantic resolvability*).

2.7 Semantic Tractability

2.7.1 Basic Concepts

Definition 2.1 (Data Model). A **data model** \mathcal{DM} is a set $\mathcal{T}_{\mathcal{DM}}$ of data model types and relations $\mathcal{R}_{\mathcal{DM}}$ between these types.

Definition 2.2 (Dataset). A **dataset** DS is a data collection which is represented under a data model \mathcal{DM} .

Definition 2.3 (Dataset Lexicon). The **dataset lexicon** Lex_{DS} of DS is a tuple of (t_0, \dots, t_n) where $t_i \in \mathcal{T}_{\mathcal{DM}}$.

Definition 2.4 (Query). A **schema-agnostic query** q can be represented by a **query** Q that is a tuple $(Token_q, Att_q)$ where $Token_q$ is the ordered set of tokens that form the question q and $Att_q : Token_q \rightarrow Token_q$ is the attachment function (syntactic relationship) between elements in $Token_q$.

Definition 2.5 (Interpretation of a Query). An interpretation of a query Q is a tuple $Q^{struct} = (E, R, L, Op, V)$, where E are a set of database elements mapped to the query, R is an ordered set of syntactic n-ary associations between elements in E , L is a set of logical operators, Op is a set of functional operators and V is a set of binding variables.

A **valid interpretation** of Q is a statement that satisfies a number of conditions (depending on the semantic model used) connecting the query tokens to the dataset lexicon.

- | | |
|---|--|
| <p>① Query: Who is the spouse of Bill Clinton?
 DB: Bill Clinton spouse Hillary Clinton
 Answer: Hillary Clinton
 Semantic Gap class: Identical
 Semantic Matching:
 <Trivial mapping, No external KB, Absolute, 1:1, Sufficient context></p> | <p>② Query: Who is the wife of Bill Clinton?
 DB: Bill Clinton spouse Hillary Clinton
 wife subPropertyOf spouse
 Answer: Hillary Clinton
 Semantic Gap class: Taxonomical
 Semantic Matching:
 <Generalization, No external KB, Absolute, 1:1, Sufficient context></p> |
| <p>③ Query: How many children does Barack Obama have?
 DB: Barack Obama child Malia Ann Obama
 Barack Obama child Natasha Obama
 Op: count
 Answer: 2
 Semantic Gap class: Aggregation/Functional
 Semantic Matching:
 <String / Functional mapping, No external KB, Absolute, 1:1, Sufficient context></p> | <p>④ Query: Is Bill Clinton married?
 DB: Bill Clinton spouse Hillary Clinton
 Answer: Yes
 Semantic Gap class: Non-taxonomical
 Semantic Matching:
 <Conceptual mapping, External KB, Absolute, 1:1, Sufficient context></p> |
| <p>⑤ Query: Give me all American presidents.
 DB: Barack Obama occupation president
 Barack Obama nationality United States
 Answer: Barack Obama
 Semantic Gap class: Predication/composition, conceptual
 Semantic Matching:
 <Conceptual, External KB, Absolute, 1:1, Sufficient context></p> | <p>⑥ Query: What is the highest mountain?
 DB: Mount Everest elevation 8848.0
 K2 elevation 8611.0
 Op: sort by desc, top most
 Answer: Mount Everest
 Semantic Gap class: Non-taxonomical, Functional
 Semantic Matching:
 <Conceptual / Functional, External KB, 1;N, Approximate / ambiguous*, Sufficient context></p> |
| <p>⑦ Query: Who are the grandchildren of Elvis Presley?
 DB: Elvis Presley children Lisa Marie Presley
 Lisa Marie Presley children Danielle Riley Keough
 Lisa Marie Presley children ...
 Answer: Danielle Riley Keough, ...
 Semantic Gap class: Non-taxonomical, Functional
 Semantic Matching:
 <Conceptual, External KB, 1:1, Approximate/ambiguous, Sufficient context></p> | <p>⑧ Query: Give me all people named James?
 DB: James Joyce name 'James Joyce'
 Answer: James Joyce
 Semantic Gap class: Lexical
 Semantic Matching:
 <Lexical, No external KB, 1:1, Sufficient context></p> |

FIGURE 2.8: Classification of existing queries according to the *lexico-semantic differences* and *semantic mappings*.

Definition 2.6 (Syntactic Mapping). Given a data model \mathcal{DM} and a query Q with interpretation Q^{struct} , a mapping function $m(Q, \mathcal{DM}) : Token_q \rightarrow E$ which defines the possible syntactic realizations of Q under \mathcal{DM} can be defined.

The syntactic interpretation of a query Q , denoted by $I(Q, \mathcal{DM})$ are the possible realizations of Q under the data model \mathcal{DM} , such that $I(Q, \mathcal{DM})$ is semantically equivalent to Q .

2.7.2 Semantic Tractability

Popescu et al. [55] defines a framework to evaluate the reliability of a NLI, formally defining the properties of *soundness* and *completeness* and identifying a class of semantic tractable natural language queries. *Semantic tractability* essentially expresses that there should be a syntactic correspondence between the syntactic structure of the query and the syntactic structure of the database.

Definition 2.7 (Semantic tractability). Given a query Q and a dataset DS with lexicon Lex_{DS} , we construct a query-dataset mapping $map_{(Q,DS)} : Token_q \rightarrow Lex_{DS}$. A query Q will be considered *semantically tractable* whenever such mapping exists.

The concept of semantic tractability assumes that there is a *one-to-one perfect synonym mapping* between the query and database lexicon which preserves the *dataset predicate-argument structure induced by the lexical categories of the query*, leaving the problem of *conceptual matching* and *more complex syntactic matching* out of the definition. This *unambiguous synonymic correspondence* which is one of the conditions for semantic tractability cannot be guaranteed in a large schema/schema-less database query scenario, where the database lexicon is potentially very large. Figure 2.9 shows an example of a natural language query and its potential corresponding and conceptual expressions over an example dataset.

Additionally, with a large vocabulary variation it is also not possible to guarantee an *syntactic correspondence between query and database*, rendering a significant part of the queries to the status of being not semantically tractable. In order to extend this classification we define the concept of semantic resolvability to cope with other category of semantic mappings.

2.8 Matching Schema-Agnostic Queries

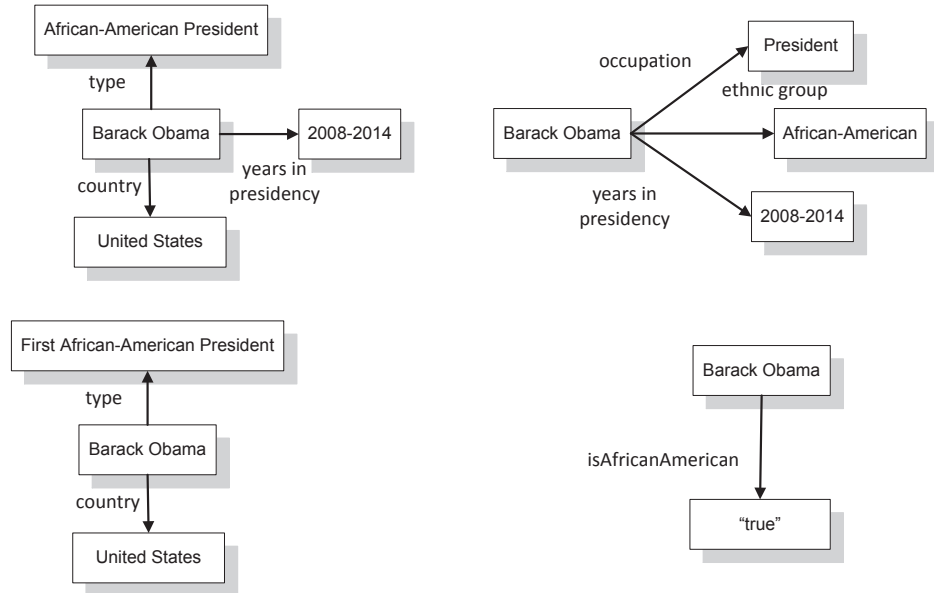
2.8.1 Semantic Resolvability

In order to define a broader class of query-dataset mappings, a *semantic Knowledge Base (KB)* which supports the $Token_q \rightarrow Lex_{DS}$ mapping is introduced.

Definition 2.8 (Semantic Knowledge Base (KB)). A *semantic knowledge base* \mathcal{M}_Σ with signature $\Sigma = (\mathcal{R}, \mathcal{E})$ is a collection of concepts constructed using two finite sets of symbols representing relations (and properties) $r \in \mathcal{R}$ and entities $e \in \mathcal{E}$.

NL Statement: Who is the first African-american president of the United States?

Possible Syntactic Mappings (RDF)



Possible Syntactic Mappings (Predicate-argument)

1: African-American President (Barack Obama) country (United States)	1-PREDICATE ₀ (CONSTANT ₀) ∧ 2-PREDICATE ₁ (CONSTANT ₀ , CONSTANT ₁)
2: First African-American President (Barack Obama) country (United States)	
3: occupation (Barack Obama, President) ∧ ethnic group (Barack Obama, African-American) ∧ years in presidency (Barack Obama, 2008-2014) ∧ country (Barack Obama, United States)	2-PREDICATE ₀ (CONSTANT ₀ , CONSTANT ₁) ∧ 2-PREDICATE ₁ (CONSTANT ₀ , CONSTANT ₂) ∧ 2-PREDICATE ₂ (CONSTANT ₀ , CONSTANT ₃) ∧ 2-PREDICATE ₃ (CONSTANT ₀ , CONSTANT ₄)

FIGURE 2.9: Example of predication differences associated with the database representation derived from a natural language statement.

Definition 2.9 (Associated Semantic KB). Given a semantic KB \mathcal{M}_Σ with signature $\Sigma = (\mathcal{R}, \mathcal{E})$ and a lexicon Lex , we say that $\mathcal{M}_{\Sigma, Lex} = (\mathcal{M}_\Sigma, f)$ is the *associated semantic KB* wrt Lex whenever f is a mapping defined by

$$f : Lex \rightarrow (\mathcal{R} \cup \mathcal{E})$$

A mapping f_{cpt} from concepts in $\mathcal{M}_{\Sigma, Lex}$ to concepts in \mathcal{M}_Σ can be defined using f as follows:

$$f_{cpt}(c(e_0, \dots, e_n)) = f(c)(f(e_0), \dots, f(e_n))$$

where $f(c) \in \mathcal{R}$ and $f(e_0), \dots, f(e_n) \in \mathcal{E}$.

Definition 2.10 (Semantic Reachability). A concept $r_n \in \mathcal{M}_\Sigma$ is reachable from a concept $r_0 \in \mathcal{M}_\Sigma$ if there is an ordered sequence $\langle r_0, r_1, \dots, r_n \rangle$ where for all $i \in [0, n-1]$, exist $u \in [1, \text{arity}(r_i)]$ and $v \in [1, \text{arity}(r_{i+1})]$ such that $\text{proj}(r_i, u) = \text{proj}(r_{i+1}, v)$ where $\text{arity}(r)$ means the arity of relation r and $\text{proj}(x, y)$ represents the y -ary argument of relation x .

A concept $c_n \in \mathcal{M}_{\Sigma, Lex}$ is reachable from a concept $c_0 \in \mathcal{M}_{\Sigma, Lex}$ whenever $f_{cpt}(c_n)$ is reachable from $f_{cpt}(c_0)$.

Definition 2.11 (Query-Dataset Semantic Mapping). Given a query Q and a dataset DS with lexicon Lex_{DS} , a query-dataset semantic mapping wrt an associated semantic KB $\mathcal{M}_{\Sigma, Token_q}$ is a mapping

$$\text{map}_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})} : Token_q \rightarrow Lex_{DS}$$

such that $\forall c \in Token_q$, if $Dep_q(c) = d$ then $f_{cpt}(d)$ is reachable from $f_{cpt}(c)$.

Definition 2.12 (Semantic Resolvability). A query Q is semantically resolvable to a dataset DS when $\forall t_i \in Token_q$ exists a semantic mapping $\text{map}_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}$ under a semantic KB \mathcal{M}_Σ which satisfies the syntactic constraints in Dep_q and DS .

Definition 2.13 (Resolved Schema-Agnostic Query). A query Q over a dataset DS is a resolved schema-agnostic query if there is a semantic KB \mathcal{M}_Σ which makes it semantically resolvable to DS .

2.8.2 Semantic Mapping Types

In the previous section the concept of *semantic mapping* was introduced without the analysis of the types and conditions involved in the semantic mappings supported by the semantic KB. However, under realistic scenarios, semantic mapping approaches will need to cope with *inconsistent*, *incomplete* semantic KBs and *ambiguous*, *vague* queries and databases. This work builds upon the basis developed in the context of schema matching (in particular adapting the work of Kashyap & Sheth [27]) to provide a classification for different types of *query-dataset mappings*.

Definition 2.14 (Semantic Mapping Type). Given a query Q , a dataset DS with lexicon Lex_{DS} and a query-dataset semantic mapping $\text{map}_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}$, for all $t_i \in Token_q$, the semantic mapping type of (t_i, e_i) , where

$e_i = \text{map}_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}(t_i)$, is defined by the tuple $(\mathcal{AP}, \mathcal{PS}, \mathcal{M}, \mathcal{SE}, \mathcal{CT}, \mathcal{MC})$, where:

1. *Abstraction Process* \mathcal{AP} : is defined as a mechanism used to map the concept associated with t_i to the concepts associated with the database elements e_i .

- (a) *Trivial*: A semantic mapping is *trivial* if the lexical expression of t_i is identical to the lexical expression of e_i and both t_i and e_i have a single word sense.
 - (b) *Lexical*: A semantic mapping is *lexical* if t_i and e_i have a *common morphological root* r .
 - (c) *Synonymic*: A semantic mapping is *synonymic* if t_i and e_i are synonyms and have the same lexical category.
 - (d) *Generalization/Specialization*:
 - i. *Generalization*: A semantic map is a *generalization* if e_i is a superclass of t_i .
 - ii. *Specialization*: A semantic map is a *specialization* if e_i is a subclass of t_i .
 - (e) *Conceptual*: A semantic map is a *conceptual mapping* if t_i and e_i are non-taxonomically related and if there is a non-taxonomical inference process supporting t_i and e_i .
 - (f) *Functional/Aggregation*: A semantic mapping is *functional* if there is a functional operator op_j which maps to t_i .
2. *Predicate Structure \mathcal{PS}* : Maps to differences in the associated predicate-argument structure from the projection of t_i into the data model \mathcal{DM} and the predicate structure of e_i .
- (a) *Predication preserving*: If the predicate-argument structure between t_i and e_i is preserved.
 - (b) *Predication difference*: If the predicate-argument structure between t_i and e_i is not preserved.
3. *Semantic Knowledge Base \mathcal{M}* : Consists of the existence of a semantic knowledge base supporting the semantic mapping.
- (a) *Self-sufficient*: The semantic mapping does not depend on a knowledge base external to the dataset.
 - (b) *Dependent on External Knowledge Base*: The semantic mapping depends on a knowledge base external to the dataset.

4. *Semantic Evidence & Uncertainty \mathcal{SE}* : Consists of the categorization of the mapping according to the supporting *semantic evidence* and *uncertainty* in the query, dataset, and in the semantic KB .
- (a) *Absolute*: A semantic mapping is *absolute* if for every possible context, the t_i maps to e_i . An absolute mapping is independent of the context provided by the query and by the dataset.
 - (b) *Context resolvable*: A semantic mapping is *context resolvable* if there is a mapping between t_i and e_i which is uniquely determined by the query and the dataset context.
 - (c) *Ambiguous*: A semantic mapping is *ambiguous* if t_i maps to different dataset elements $e_i \cdots e_{i+n}$, where $e_i \cdots e_{i+n}$ have meanings which do not generate valid interpretations for the query.
5. *Context \mathcal{CT}* : Consists of the query context $Q^{context} = \{t_i \mid t_i \in Token_q\}$ and the dataset context $DS^{Context} = \{e_i \mid e_i = map_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}(t_i)\}$
- (a) *Sufficient*: The context is *sufficient* to determine the query-dataset mapping given context-resolvable semantic evidence scenario.
 - (b) *Insufficient*: The context is *insufficient* to determine the query-dataset mapping given context-resolvable semantic evidence scenario, leading to ambiguity or vagueness in the query-dataset semantic mapping.
6. *Mapping cardinality \mathcal{MC}* :
- (a) *Single mapping (1 : 1)*: A semantic mapping is a *single mapping* if $map_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}$ is a one-to-one map.
 - (b) *Data redundant (1 : N)*: A semantic mapping is *data redundant* if $map_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}$ is a multi-valued map.
 - (c) *Query redundant (N : 1)*: A semantic mapping is *query redundant* if $map_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}$ is a many-to-one map between $Token_q$ and DS .
 - (d) *Query-data redundant (M : N)*: A semantic mapping is *query-data redundant* if $map_{(Q, DS, \mathcal{M}_{\Sigma, Token_q})}$ is a many-to-many relationship between $Token_q$ and DS .

The concept of *semantic tractability* corresponds to the tuple $(\mathcal{AP} = \{*\}, \mathcal{PS} = \{\text{Predication Preserving}\}, \mathcal{M}, \mathcal{SE} = \{\text{Absolute, Context Resolvable}\}, \mathcal{CT} = \{\text{Sufficient}\}, \mathcal{MC} = 1 : 1)$,

	Abstraction Process	Predicate Structure	Mapping Cardinality	Semantic Evidence & Uncertainty	Semantic Knowledge Base	Context
↑ Semantic resolvability	Trivial	Structure preserving	1:1	Absolute	Self Sufficient	Sufficient
	Lexical		1:N			
	Synonymic		N:1	Context resolvable		
	Generalization/ Specialization					
	Conceptual	Structure difference	M:N	Ambiguous	Dependent on External KB	Insufficient

FIGURE 2.10: Semantic resolvability for different mapping types.

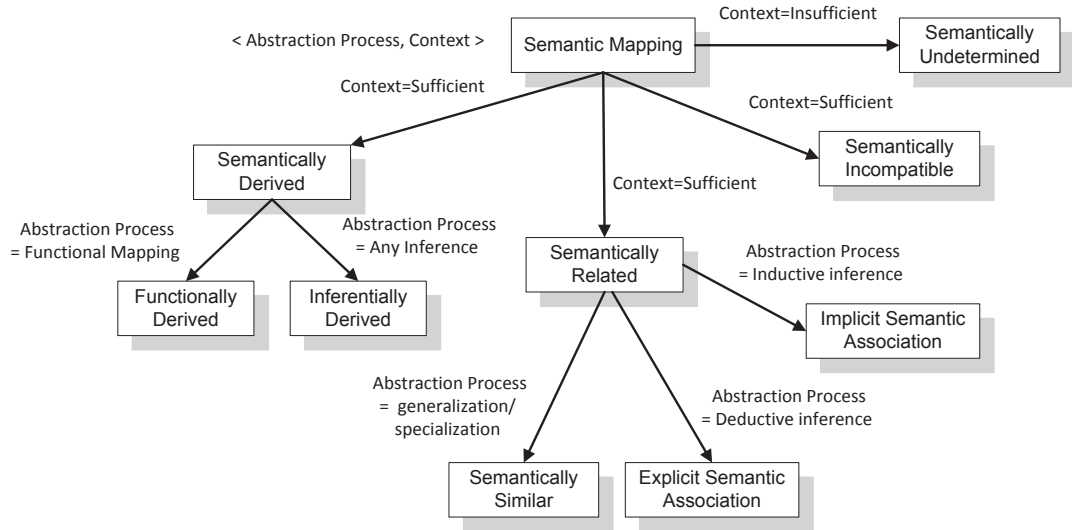


FIGURE 2.11: Semantic relationships between different semantic mapping configurations for abstraction and context (adapted from Kashyap & Sheth [27]).

which corresponds to a small subset of the possible mapping types. Figure 2.10 shows levels of semantic resolvability for different mapping types and Figure 2.11 shows the *semantic relationships* for different abstraction-level and contexts.

The process of assigning a database associated interpretation $I_{DS}(Q)$ to a schema-agnostic query Q depends on coping with the semantic phenomena of *term ambiguity*, *structural ambiguity*, *vagueness* and *synonymy*, given the query Q , the dataset DS and the semantic KB \mathcal{M}_Σ .

2.8.3 Discussion

In this section we provided a preliminary framework for modelling the semantic differences and the types of semantic mapping between schema-agnostic queries and structured databases. We generalised the semantic tractability framework proposed by Popescu et al. [55] in two directions: (i) proposing a model which is data model independent (in contrast with the relational focus on relational databases present in [55]) and (ii) deriving a new set of categories for classifying query-database mappings. We argue that the concept of semantic tractability maps to just a small subset of the possible query-database mapping conditions leaving most of the types of schema-agnostic queries out of the discussion. This work aims at providing a more comprehensive classification framework based on the concepts of *semantic resolvability* and *mapping types*.

2.9 Chapter Summary

This chapter concentrates on the analysis of the changes in the database landscape, motivating how the growth in size, complexity, dynamicity and decentralisation of schemas (SCoDD) are bringing fundamental demands for contemporary data management. These demands strongly impact the effectiveness of existing approaches for querying large-schema, semantically heterogeneous structured data. At the center of this problem is the concept of semantic heterogeneity between query and databases, which defines the vocabulary problem for databases. The dimensions and causes of query-dataset semantic heterogeneity were analysed and adapted from previous literature work.

While the understanding of the motivation for schema-agnostic queries is progressively becoming a known concern, there is a lack of categorization to express different semantic challenges that a schema-agnostic query mechanism need to cope with. In order to address this gap, this chapter introduced a classification system based on *semantic mapping categories*, which define the degree of *semantic resolvability* of a schema-agnostic query, i.e. the level of complexity involved in mapping a schema-agnostic query to a database. The goal is to provide an initial classification framework which could both help in the understanding of the challenges of schema-agnostic queries and on the scoping in the evaluation of existing approaches.

Chapter 3

Literature Review

3.1 Introduction

This chapter provides an overview of different approaches for flexible querying and searching over structured data. This analysis aims at providing an overview of the problem from the perspective of different research areas (Information Retrieval, Natural Language Processing, Semantic Web and Databases). Since the problem of querying large-schema/heterogeneous datasets is strongly emphasized in the Semantic Web/Linked Data literature, this analysis concentrates on the description of query and search approaches under this scenario.

The set of works analyzed in the state-of-the-art query approaches for Semantic Web/Linked Data were selected based on their relevance for the discussion. Four subcategories of query approaches were mapped: vector search models, approximate queries for Semantic Web/Linked Data datasets, natural language queries over Semantic Web/Linked Data datasets and visual query interfaces for Semantic Web/Linked Data datasets. Each analyzed work contains a description of the key features of the approach, a description of its evaluation and a critical analysis based on the coverage of the core requirements for schema-agnostic queries. The results of the state-of-the-art analysis are summarized in Section 3.8, where the existing gaps in the literature in terms of research and evaluation methodologies are described.

While natural language interfaces for Semantic Web/Linked Datasets have concentrated more on addressing the semantic gap (a central concern for schema-agnostic queries) other approaches have also introduced techniques for schema-agnostic queries, focusing on complementary concerns such as performance and scalability. Additional literature analysis is covered on Chapter 10, where uses of distributional semantic models in the context of structured data are analysed.

3.2 Requirements

This work focuses its contribution on the provision of a solution for addressing schema-agnostic queries. This section provides a set of requirements for schema-agnostic queries based on implicit and explicit requirements present in the literature. The list of requirements will be used in the analysis of the contribution and completeness of the proposed query mechanism and in the analysis of related work.

This investigation uses three key references in the literature to support the list of requirements. Stuckenschmidt & van Harmelen [35], describe the key elements necessary for query processing on the Semantic Web. According to Stuckenschmidt [93], conventional database techniques are not sufficient to address the challenges involved in querying the Semantic Web, and a set of five key elements (*approximation*, *integration*, *discovery*, *deduction* and *transformation*) are introduced as critical features that need to be present in a query mechanism for the Semantic Web.

Lopez et al. [42] analyses the limitations of the current approaches for interacting with Linked Data. Lopez et al. introduce a set of six key criteria that are used for analyzing paradigms to interact with the Web of Data: *usability*, *expressivity*, *scalability*, *mapping*, *fusion* and *ranking*. The third work analyzed is the informal set of requirements introduced by Wang et al. [31] in the design of Semplore, an approximate query mechanism for Linked Data. Wang introduces four requirements: *imprecise information needs*, *usability*, *scalability* and *data change (timeliness)*.

Complementing this initial set of requirements based on the literature, additional requirements are introduced in order to provide a complete picture of the set of features that need to be addressed by a schema-agnostic and expressive natural language query mechanism for Linked Data. The set of additional requirements were collected from generic and common requirements for search engines (performance/scalability, accuracy/completeness). Table 3.2 shows the requirements and their associated mappings in the literature.

Requirements	Stuckenschmidt [93]	Lopez et al. [42]	Wang et al. [31]
Accurate & comprehensive semantic matching	approximation, transformation, deduction, integration, discovery	mapping, fusion, ranking	imprecise information needs & disambiguation mechanisms
High query expressivity		expressivity	
High usability & Low query construction time		usability	usability
Ability to query distributed datasets	integration	fusion	
Interactive search & Low query-execution time		usability	usability
Low setup & maintainability effort		usability	
High scalability			
Dataset Discovery	discovery		
Timeliness of the data			data change

TABLE 3.1: Requirements dimensions and their correspondence on literature.

Since this work concentrates on the semantic matching dimension for schema-agnostic queries, the requirements *timeliness of the data*, *ability to query distributed datasets* and *dataset discovery* are left out of the analysis scope of this work.

The final list of *requirements for schema-agnostic queries* are:

1. *High usability & Low query construction time*: Support for a simple and intuitive interface for experts and casual users. Reference: usability in [42], [31].
2. *High expressivity*: Queries referencing *structural elements and constraints* in the dataset (relationships, paths) should be supported, as well as *operations* over the data (e.g. aggregations, conditions). Reference: expressivity in [42] and the query expressivity in structured query mechanisms.
3. *Accurate & comprehensive semantic matching*: Ability to provide a principled semantic matching addressing all the dimensions of the semantic heterogeneity problem (abstraction, conceptual, compositional, functional). Semantic matching with high precision and recall. Reference: approximation and transformation in [93], mapping and ranking in [42], imprecise information needs [31].
4. *Low setup & maintainability effort*: Easily transportable across datasets without significant manual adaptation effort. The query mechanism should be able to work under an open domain and across multiple domains. Databases should be indexed with a minimum level of manual adaptations. Minimization of user intervention in the construction of supporting semantic resources used in the semantic matching. Reference: usability in [31, 42].
5. *Interactive search & Low query-execution time*: Minimization of user interaction/feedback effort in the query process. Users should get answers with interactive response times¹ for most of the queries. Reference: usability in [31, 42].
6. *High scalability*: The query approach should scale to large datasets both in query execution and indexing construction time. The query approach should scale to a large number of datasets. Reference: scalability in [31, 42].

3.3 Approaches for Querying Semantic Web/Linked Data Datasets

Different approaches have been investigated in the process of searching and querying Semantic Web/Linked Data datasets. Current techniques range from the application of

¹an interactive query execution time is contrasted with a batch query execution time

classical vector search models widely used under the information retrieval perspective to approximation/constraint relaxation techniques applied to structured queries over Semantic Web/Linked Data datasets.

This work categorizes existing approaches into four categories:

- *Entity Search: Vector Space models for Semantic Web/Linked Data datasets;*
- *Approximate queries for Semantic Web/Linked Data datasets;*
- *Natural language queries for Semantic Web/Linked Data datasets;*
- *Visual query interfaces for Semantic Web/Linked Data datasets;*

The sections below describes existing works in each category.

3.4 Entity Search: Vector Space Models for Semantic Web/Linked Data Datasets

Strategies inherited from the information retrieval space for unstructured data are used in the creation of search engines for Semantic Web/Linked Data. The commonality across these approaches are the use of vector space models/inverted indexes to search over structured data. Most of the evaluation of entity search approaches focuses on the measurement of query execution/indexing time. More recently, test collections such as the Semantic Search Challenge [78] have been used to evaluate the quality of the results of existing approaches. From the user interaction perspective entity search varies from keyword-based search queries to hybrid keyword structured queries. In the rest of this section, a set of prominent approaches under this category are described.

3.4.1 Sindice/SIREn (Tummarello et al. [94])

Description: Sindice [94] focuses on the provision of an entity-search service for Linked Data. Instead of focusing on complex structured queries, Sindice focuses on an entity-centric approach, where the search is targeted towards the retrieval individual instances, classes and properties in the Linked Data Web. The approach used in SIREn, the search engine behind Sindice, combines query-dependent and query-independent ranking techniques to compute the final entity scores. The query-dependent score function uses a variation of the TF-IDF weighting scheme (*term frequency - inverse subject frequency*: TF-ISF) to evaluate individuals by aggregating the values of partial scores for predicates

and objects. The TF-ISF scheme gives a low weight to less discriminative keywords associated with entities. Additionally, the query-independent approach uses *hierarchical link analysis* [95, 96] to rank entities using a two layered model of the Linked Data Web, where the top layer is represented by datasets and their interlinking, and the lower layer is composed of interlinked entities. The hierarchical random walk model takes into account both dataset and entity level interlinking to compute the query-independent rank score. SIREn supports three types of queries: *keyword-based search* (which abstracts users from the structure of the data), *semi-structured star-shaped queries* (where elements surrounding an entity are referred in the query: in this case the user depends on the knowledge of the vocabulary behind the dataset), and a combination of the two previous approaches, where users can query with a partial description of the data model.

Evaluation: SIREn is evaluated in two directions: in relation to the performance and scalability of the indexing process and in relation to the performance and accuracy in the search and ranking process. In the indexing evaluation part [95], Delbru et al. measures the *index size*, *insertion time*, *query execution time* and *scalability*. In these experiments SIREn is compared against triple stores (RDF-3X and Sesame) as baselines. Two datasets are used: one generated from a random sampling of Sindice crawled data (generating a final dataset containing 10M triples) and the MIT Barton Dataset [97]. On the search and ranking side [95, 96], Delbru et al. conduct two experiments using as the baseline algorithm a global version of EntityRank [95]. The first experiment focuses on the investigation of the impact of link locality in the ranking process. The objective of this experiment is to verify if the local approximation introduced in the proposed ranking approach represents a good approximation to the global approach. In this experiment DBpedia, Citeseer and Geonames are used. The second experiment performs a user study to measure the effectiveness of the search algorithm, where users are asked to evaluate the ranking effectiveness of the three ranking strategies by performing a well defined search task.

Critique: The SIREn indexing approach is based on an adaptation of the TF-IDF vector search models, under a Lucene-based² inverted index infrastructure, providing a scalable, high performance ranking solution for entity search. SIREn is not designed to address complex structured queries and does not fully cover the query expressivity requirement. The main search interface of SIREn is keyword-based but it does not provide full expressive natural language queries. SIREn does not support expressive schema-agnostic structured queries (e.g. coping with semantic differences between query and vocabularies terms).

²[urlhttp://lucene.apache.org](http://lucene.apache.org)

3.4.2 SWSE (Harth et al. [33])

Description: SWSE is an entity-centric Semantic Web search engine, concentrating on the analysis of a full architecture/data management infrastructure for a search engine for the Semantic Web. The SWSE architecture includes components to crawl, integrate, transform, enhance, index, query and navigate over multiple data sources. The main components in the architecture of the system consist of *query processing*, *ranking*, *indexing manager* components and an internal quad store (YARS2). YARS2 focuses on scalability issues to enable federated queries over Web data. ReConRank [98] is the approach used for ranking entities and consists of a contextual adaptation of the PageRank/HITS algorithm to RDF data. OWL reasoning is provided by SAOR [99].

Evaluation: The evaluation of SWSE, its data repository (YARS2) and ReConRank focuses on performance and scalability issues. In the case of SWSE, no quality of the ranking results is performed. For the performance evaluation of ReConRank, 2.8GB of data crawled from the Web was used for measuring the ranking computation time. The experimental evaluation of YARS2 [100] focuses on the scalability features of the system, including the evaluation of distributed join performance for variable result set sizes.

Critique: SWSE focuses on the provision of two types of services: *keyword-based entity search* and *SPARQL queries* over data collected and integrated from the Web. The main strength of the work is its focus on the provision of a high-performance and scalable solution and the provision of an integrated architecture. The system does not target schema-agnosticism: users need to be aware of the vocabularies behind the data to issue complex queries over the aggregated data. Complex queries can only be executed using SPARQL (low usability). The quality of the entity ranking approach is not evaluated. SAOR is the OWL reasoning component of SWSE.

3.4.3 Semplore (Wang et al. [31])

Description: Semplore [31] is a search engine for Linked Data which uses a hybrid query formalism, combining keyword search with structured queries. The system covers different aspects of a solution for searching and querying Linked Data, including the management of index updates and the use of user interaction through facets. The Semplore approach consists in indexing the entities of the Linked Data Web using the associated tokens and sub/superclasses as indexing terms. In addition to entity indexing, Semplore focuses on indexing relations using a position-based index approach (PosIdx: position based index) to index relations and join triples. In an inverted index, each term is associated with a posting list of documents. For each of the documents there is

a position list showing the position where the terms appear. In the PosIdx approach, relation names are indexed as terms, subjects are stored as documents and the objects of a relation are stored in the position list. For example, for a triple (s, p, o) , the object o is stored in the position list of the term p in the document s . Based on the proposed index, Semplore reuses the IR engine's *merge-sort based boolean query evaluation method* and extends it to answer unary tree-shaped queries.

Evaluation: Semplore uses the LUBM [101] benchmark datasets for the evaluation of its scalability. DBpedia [74] and YAGO [102] data is used in the evaluation of query expressivity and performance. Query expressivity is evaluated by classifying the queries into 5 query categories: single-atom queries, path queries, star-shaped queries, entity queries and tree-shaped queries. These queries categories serve as the base for the evaluation of the query execution time of the approach. The evaluation of the precision and recall of the approach is limited. For this purpose, four query datasets are created: QS1 contains 500 queries based on simple keyword-search; QS2 contains 1.242 queries covering 621 simple relations in DBpedia; QS3 is composed of 287 queries involving three constant relations and QS4 involves the manual creation of 20 queries based on questions provided by 10 users.

Critique: Semplore focuses in the provision a scalable approach for expressive queries for Linked Data. The core contribution of Semplore is the introduction of an indexing mechanism for relations which allows the answering of complex boolean queries. The Semplore approach is based on a VSM, being highly scalable and showing both high performance for indexing and query execution. Query expressivity is evaluated under a simple categorization, addressed with the paradigm of SPARQL queries in mind. The concern with schema-agnosticism is partially covered by the introduction of ontology based taxonomic enrichment of the terms in the index. The system is not evaluated in terms of precision/recall under a schema-agnostic scenario. OWL reasoning is applied before loading the data into the index, which provides the second mechanism for semantic enrichment. The approach intrinsically copes with the ability to query distributed datasets once datasets present on the Web are indexed.

3.4.3.1 Dong & Halevy [103]

Description: Dong & Halevy propose an approach for indexing dataspace [103] allowing queries that combine keywords and structural relations. The index structure is designed to cope with two query types: predicate queries and neighborhood keyword

queries. The first type of queries cover conjunctions of predicates and associated keywords (e.g. (title 'Birche'), (author 'Raghu'), (publishedIn '1996 Sigmod')) while the second type covers keyword searches over interrelated instances.

Dong & Halevy introduce four structured index types: attribute inverted lists (ATIL), *attribute-association inverted lists* (AAIL), *attribute inverted lists with duplication* (Dup-ATIL), *attribute inverted lists with hierarchies* (Hier-ATIL) and *hybrid attribute inverted lists* (Hybrid-ATIL). All approaches are based on the introduction of additional structure information as concatenated terms in the inverted lists. Taxonomy terms are introduced in the index using the same strategy. Schema-level synonyms are handled using synonyms tables.

Evaluation: The approach is evaluated using the following measures: *query execution time* (for 1 query clause, 2 query clauses, 5 query clauses), *index look-up time*, *indexing time*, *index update time*, and *scalability*. The evaluation dataset was built from personal information management data and external data sources. The final dataset contained (105,320 instances and 468,402 triples). The approach is not evaluated in terms of quality of results (precision/recall).

Critique: The approach proposes a flexible and structured query mechanism over datas-paces by introducing structural information of the data in the index inverted lists. The main limitations of the work include the lack of a ranking mechanism and the absence of an evaluation of the relevance of the results (precision/recall). Despite the discourse on the vision of coping with data heterogeneity, the proposed inclusion of relationships on the index in itself is not a comprehensive solution for schema-agnosticism, where the discussion on semantic matching concentrates on synonymic index expansion.

3.4.3.2 SPARK (Zhou et al. [104])

Description: Zhou et al. [104] proposes SPARK, an approach which translates keyword-based queries to SPARQL. The SPARK approach is based on three basic steps: *term mapping*, *query graph construction* and *query ranking*. The term mapping step consists in finding the elements in the ontology for each term in the keyword query. The names and the terms in the ontology are used for mapping. The approach uses two types of mapping: (1) morphological mapping, employing stemming, edit distance and substring matching techniques and (2) semantic mapping using synonyms provided by WordNet. The second step, query graph construction builds up the candidate query graphs with the ontology resources previously mapped. The mapped elements are split into different query sets and a *minimum spanning tree algorithm* is applied to build possible query graph patterns. The term mapping and the query graph construction steps outputs

multiple candidate queries based on the original keyword query. A *probabilistic query ranking* approach is introduced as a strategy to rank possible queries. The probabilistic ranking model consists of two sub-models. The first one, the *Keyword Query Model* (KQM), takes into account the probabilistic relation between the keyword query terms and the formal query. The second, *Knowledge Base Model* (KBM), measures the information content of each formal query, ranking higher query configurations with higher information content (lower probability). According to the experiments the KQM part provided a more consistent contribution compared to the KBM part. KBM however helped smoothing the degradation of KQM with the increase of the query length. The final ranking model, the combination of KQM and KBM, strongly degrades for query lengths greater than six.

Evaluation: In the evaluation of SPARK, Zhou et al. manually construct the query and the dataset from the Tang & Mooney Natural Language Learning Data [67]. Zhou et al. convert the original datasets (covering the geography, job and restaurant domains) into RDF/S ontologies. Instead of using full natural language queries provided by the original dataset, the authors converted each query into keyword-like queries. The final translation done by SPARK from the keyword query into SPARQL is compared with the correct SPARQL, if the two queries are not semantically equivalent it is considered as a false result. Zhou et al. uses two metrics in the evaluation: *mean reciprocal rank* (MRR) and *recall*. 250 keyword queries were generated for each dataset and the variation of MRR against the query length was measured. Additional metrics such as *average keyword query length* and *query processing time* were collected. The authors also describe a user study based on the measurement of user satisfaction of 50 online users.

Critique: The SPARK approach provides a ranking solution for translating keyword-based queries to low complexity SPARQL queries, targeting low complexity RDF datasets. The work provides an evaluation consistent with this scenario, using the test collection based on the Tang & Mooney[67]. The matching process between query terms and ontology terms is limited to WordNet synonyms, showing a limited solution to address schema-agnostic queries. The solution provides keyword-based queries and partially addresses the ability to cope with more expressive queries. More complex query operations (e.g. aggregation), however, are not available, making the query expressivity of the solution limited. Additionally, the scalability and the performance of the proposed solution in a Web scale scenario were not evaluated.

3.5 Approximate Queries for Semantic Web/Linked Data Datasets

This section describes different approaches for approximate query mechanisms for Semantic Web/Linked Data datasets. Approximation strategies vary from the relaxation of query constraints to the application of similarity functions over structured data.

3.5.1 Oren et al. [105]

Description: Oren et al. [105] proposes an approximate and evolutionary approach for querying RDF data. *Bloom filters* are used for improving the performance of the approximation process. The approximation is done by relaxing the constraints and progressively restoring them through an evolutionary approach. The matching is done by progressively evolving the solutions using standard evolutionary methods over the application of the query graph constraints and by measuring the resulting fitness function where the quality of the results in the query process increases monotonically. The evolutionary process consists in the application of four operators (*parent selection*, *crossover*, *mutation* and *survivor selection*) in the calculation of a fitness function.

Evaluation: The approach is evaluated using the three datasets: LUBM (which is targeted towards OWL reasoning), a collection of publicly available FOAF profiles and extraction from DBLP dataset. One query is evaluated for each dataset. For the DBLP datasets, one query of the benchmark queries proposed by Svihla & Jelinek [106]; for LUBM the LUBM query #2 was used. The experiments measured *data loading time*, *query execution time*, and *average and best fitness of the population*.

Critique: The proposed approach executes approximate queries using an evolutionary approach over the query constraints. The main motivation of the approach is approximation for query performance. Users need to be aware of a partial structure of the vocabularies: SPARQL is used to interact with the data, where expressive queries are supported in the context of SPARQL. The approach showed low query execution times with the datasets evaluated.

3.5.2 Stuckenschmidt & van Harmelen [35]

Description: Stuckenschmidt & van Harmelen [35] approach the problem of approximate reasoning by relaxing all the query constraints and iteratively restoring them, progressively refining the result set. The set of proposed relaxation strategies are based on the structure of the query graph as defined in Horrocks & Tessaris [107]. A 'good'

approximation is defined by the monotonic increase in the quality of the results in every step of the approximation. The creation of good approximation strategies relies on the fact that further query constraints should be applicable over the current set of objects present in the result set. The underlying assumption is that simpler conjunctive queries can be answered in shorter time. The approximation steps are based on uninformed search strategies over the query graph. Two types of strategies are investigated: node expansion approximations and arc-based approximations.

Evaluation: The proposed approach was analysed in a theoretical context.

Critique: In [35] the approach is described in theoretical terms and it is not implemented or evaluated. The analysis of the strengths and limitations of the approach is limited due to the lack of experimental analysis. The nature of the approximation is purely constraint-based, does not fully address the complexity of the semantic matching to support schema-agnostic queries. In this scenario, users still need to be partially aware of the vocabulary of the data.

3.5.3 Hurtado et al. [36]

Description: In [36], Hurtado et al. investigate a flexible query mechanism for RDF data, with the purpose of supporting varying degrees of exactness in RDF queries. The approach consists in the application of a logical relaxation of the query constraints based on RDFS, returning a ranked set of results. As a motivation for this work, Hurtado et al. focus on the scenario where there is a lack of understanding of the ontology behind the data. Differently from the relaxation allowed in SPARQL through OPTIONAL, where triple pattern constraints are eliminated, the approach focuses on a logical relaxation, where class and property hierarchies in the ontology associated with the data could be used in the relaxation of the constraints. In this approach, a triple pattern could be relaxed by navigating over the classes and properties hierarchies of the triple (*rdfs:subclass*, *rdfs:subproperty*, *rdfs:type*). The authors propose an extension of the SPARQL language to support the RELAX clause, where the logical constraints of a triple under the scope of this clause could be relaxed. A complementary type of relaxation based on RDFS entailment is also considered, which includes removing triple patterns, replacing constants with variables and breaking join dependencies.

Evaluation: The proposed approach was analysed in a theoretical context.

Critique: The approximation model is based on logical relaxation: the approach relies on the fact that there is sufficient taxonomical RDF data to support the approximation

process. Additionally the semantic approximation model is strongly based on taxonomic relaxation.

3.5.4 SPARQLer (Kochut et al. [108])

Description: SPARQLer [108] is a SPARQL extension which allows query and retrieval of semantic associations (complex relationships) in RDF. The SPARQLer approach is based on the concept of path queries where users can specify graph path patterns, using regular expressions for example. The pattern matching process has been implemented as a hybrid of a bidirectional breadth-first search (BFS) and a simulation of a deterministic finite state automaton (DFA) created for a given path expression. For each instance of the iterator created for a path pattern, two DFAs are constructed. The first DFA recognizes the regular language defined by the original path expression and the second recognizes the reversed language. The search process uses bidirectional BFS to iteratively grow candidate paths. A candidate path is generated when an entity on the leaf nodes of the bidirectional search matches. The objective of SPARQLer is to provide a solution targeted towards the query of semantic associations, where some path pattern between two entities can be expressed using regular expressions. SPARQLer is included in this literature review due to the possibility of using the approach to support semantic approximation and approximate path queries.

Evaluation: SPARQLer is evaluated using a modified version of the DBLP dataset and using the GlycO ontology (362 classes and 84 properties). The first dataset is used to evaluate the practical scalability and performance of the approach.

Critique: SPARQLer is used as a query approach in a more specific scenario, where path queries between two entities are approximated by the definition of a regular expression. Regular expressions allow an approximate description of paths between two entities. Using regular expressions users can specify flexible path constraints in terms of the relative position and occurrence of the graph entities. However, terms used in the query need to match exactly to the terms in the vocabulary. The approach provides an expressive query mechanism for the proposed problem by extending SPARQL.

3.5.5 iSPARQL (Kiefer et al. [37])

Description: Kiefer et al. [37] introduce iSPARQL, a similarity join extension to SPARQL, which uses user-specified similarity functions (*Levehnstein*, *Jaccard* and *TF/IDF*) for potential assignments during query answering. iSPARQL focuses on the problem of integrating different schemas using similarity-based joins. Kiefer et al. considers that the

choice of a best performing similarity measure is context and data dependent [37]. Three approaches were considered for extending SPARQL: *virtual triples*, *extension functions* and *solution modifiers*. The virtual triples approach call customized similarity functions under the IMPRECISE extension query block. The *extension function* uses existing SPARQL filters combined with customized similarity functions and the *solution modifier* approach extends the official SPARQL grammar with new solution modifiers. iSPARQL focuses on the provision of a similarity-based extension to SPARQL. By analyzing the pros and cons of the three approaches, Kiefer et al. conclude that virtual triples are superior to the other approaches.

Evaluation: The proposed model was applied to a data integration experiment over two RDF datasets and an ontology mapping experiment. Both scenarios used SwetoD-*blp* [109] (containing bibliographic information of Computer Science publications) The semantic data integration experiment was targeted to the determination of the applicability of iSPARQL to Semantic Web data integration. The accuracy of one query related to the integration between the two datasets was analyzed. In the second experiment, the defined task was to discover classes in different ontologies describing the same concepts.

Critique: iSPARQL provides an approach to introduce user defined similarity in SPARQL queries. By delegating the definition of a similarity model to end users, the approach introduces additional complexity in the query process. The solution can introduce some level of semantic flexibility in the query process. Some schema constraints such as the ability to map multiple properties into one property are not addressed. In addition, the work lacks depth in the investigation and evaluation on the effectiveness of the similarity measures used. Performance measures are not available.

3.6 Natural Language Interfaces for Semantic Web/Linked Data Datasets

This section covers existing approaches based on natural language queries for Semantic Web/Linked Data.

3.6.1 NLP-Reduce (Kaufmann et al. [110])

Description: NLP-Reduce is a natural language interface for querying Semantic Web knowledge bases [10]. The NLP-Reduce approach starts by pre-processing the target Semantic Web dataset by extracting and by doing synonym expansion for the triples in the dataset. After the user inputs a query, the input query processor component

pre-processes the query by removing stop-words and stemming the remaining words of the input query. The set of stemmed words are then passed to the query generator component, which attempts to match the query words to the synonym-enhanced triples, generating SPARQL queries for the matches. The SPARQL generation process starts by matching query terms to the properties of the triples present in the expanded representation. The system ranks the matching properties considering the string similarity (edit distance) of the stems and the number of matching terms (for multi-word properties). NLP-reduce then searches for data properties that matches the remaining words. In the matching process the properties domains and ranges are used. The triples with the highest scores are joined in a SPARQL query graph pattern.

Evaluation: In [10] Kauffman evaluated the performance of NLP-Reduce, using an OWL translation of the Tang & Mooney dataset [67]. 251 queries over the restaurant dataset and 879 queries over the geography dataset were executed and precision and recall were measured. In [10], Kauffman extends the evaluation of the system, doing a standard SUS usability study and comparing the results of precision, recall and usability metrics against other query mechanisms.

Critique: NLP-reduce processes the queries as bag-of-words, not exploring the syntactic structure of the sentences. The query structure is generated by the relationships of the elements in the dataset, not using the structure of the query. The solution partially addresses the provision of a schema-agnostic solution: however the matching process between query terms and conceptual model terms does not provide an effective semantic matching solution, being based on WordNet-based synonyms matching. The evaluation of the system done by Kauffman et al. showed medium-to-high precision and recall in a scenario constrained to datasets with a simple structure/smaller schemas. Kauffman et al. do not provide an evaluation of the execution performance of the system. The approach does not introduce scalability strategies (such as indexing). The system also does not cope with the ability to query distributed datasets, since it generates one SPARQL query for each dataset, not joining the answers. Reasoning is done at the dataset level and it is not incorporated into the query process.

3.6.2 Querix (Kaufmann et al. [111])

Description: Querix [111] is a domain-independent natural language interface for querying ontologies that uses clarification dialogs to eliminate ambiguities in the query process. The approach consists of seven components: a *user interface*, an *ontology manager*, a *query analyzer*, a *matching center*, a *query generator*, a *dialog component*, and an *ontology access layer*. The user interface allows users to enter natural language queries

and choose the dataset which will be queried, displaying the generated SPARQL query and the returned results after the query is executed. The ontology manager component loads user ontologies in the system enriching them with synonyms derived from WordNet. In the query analyzer component, the query is parsed using the Stanford Parser [112], where the lexical categories and the *C-structure* of the query are generated. From these two inputs a sequential structure (*query skeleton*) is generated and each noun and verb in the structure is enriched with WordNet synonyms. The matching center attempts to match the query skeleton with the synonym-enriched ontology description. The matching process starts filtering triple patterns (subject-predicate-object) in the query. After this step, the matching center searches for all potential matches between the synonym-enriched nouns and verbs present in the input query with the synonym-enriched terms in the ontology (including domains and ranges). The matching center then uses the term matching candidates to match the triple patterns identified in the query skeleton to the ontology. The query generator then joins the identified triples and creates a SPARQL query with the ontology terms. In case the query generator detects ambiguities (different possible query solutions), it prompts users with a *disambiguation dialog component*, where query alternatives are displayed. The ontology access layer component is provided by the Jena framework where the final SPARQL query is executed.

Evaluation: Kauffman et al. evaluates Querix in terms of precision and recall by using an OWL translation of the Prolog database of United States geographical facts (Tang & Mooney [99]). The translated OWL knowledge base contains 9 classes, 11 datatype properties, 17 object properties, and 697 instances. A set of 877 natural language questions which were composed from real usage of the Web interface are part of the same dataset. The approach achieved an average recall of 87.11% and precision of 86.08% for the dataset of queries addressed. Neither Kauffman nor Tang & Mooney provide a categorization of the semantic matching process present in the dataset, limiting the interpretation of the results. The small size schema of the ontology also limits the dataset for a schema-agnostic query scenario. Kauffman also did not evaluate Querix in terms of scalability and query execution time performance.

Critique: Compared to NLP-Reduce, the system introduces two main features: a disambiguation component and the application of syntax analysis over the query input. The user feedback disambiguation component is one important element in natural language based systems, where the ambiguities inherent to the natural language can negatively impact the precision of the system. The syntax analysis component also brings an initial structure to the graph pattern of the generated SPARQL query, reducing the number of potential false matches. As in NLP-Reduce, the query system focuses on querying one data source at time. There is no discussion on the scalability of the matching strategy.

The matching process also relies on WordNet synonyms to introduce semantic flexibility. The approach provides a full natural language query interface where users are not constrained in the query process. Query expressivity is not fully evaluated within the query set, which is not categorized and analyzed. Deductive DL reasoning is done.

3.6.3 Ginseng (Bernstein et al. [113])

Description: Ginseng [113] is a guided input (controlled) natural language query mechanism to an OWL knowledge base. Ginseng users query by using a closed vocabulary derived from the ontologies which are currently loaded in the system. The terms in the loaded ontologies can be enriched with annotated synonyms. The system has a query autocomplete function on the interface: users start typing the query and a list of suggested terms allowing a valid query are displayed. Ginseng is composed of four main components: *a grammar compiler, a multi-level grammar, an incremental parser and an ontology access layer* [113]. The ontology access layer is provided by the Jena framework [114]. When an ontology is loaded, the set of ontology-independent static grammar rules (defined by generic sentence structures) are extended with rules derived from the ontology. The final grammar is used to parse the input query, to provide valid *autocomplete suggestions* and to help building SPARQL queries. The incremental parser keeps an in-memory structure containing all parse paths of the current state of the input query. After the final user query is typed, a simple transformation from the parse path to the SPARQL query is made. In the parsing process it is possible to have multiple parse trees for a query. In this case Ginseng performs multiple SPARQL queries and return to the user the union of all result sets.

Evaluation: In [113], Ginseng is evaluated through a user study using the geographic dataset proposed in Tang & Mooney [67]. 20 individuals were asked to enter 30 queries over the geographical dataset. The individuals were asked to formulate natural language queries for randomly selected queries descriptions using Ginseng and the other half using SQL. After each session a *system usability scale* (SUS) [115] standardized usability test was performed, collecting the individuals impressions about the experience. The query formulation time, precision and recall were measured with the set of queries that Ginseng could parse correctly from the overall 880 queries of the Tang & Mooney geographical dataset. In [116], Kauffman evaluates Ginseng comparatively against other query mechanisms using a similar evaluation approach.

Critique: The main mechanism used by the approach to cross the semantic gap between users and knowledge bases is based on constraining the number of vocabulary terms and the query variations allowed in the autocomplete mechanism (based on controlled

natural language). In the evaluation of the system, users are faced with a small number of small sized ontologies. The approach may fail to scale for scenarios involving a large number of large sized ontologies. The constraining of the input vocabulary and the use of ontology-based grammars brings a simplification in the process of building the SPARQL queries. However, the approach is limited in terms of scalability since it relies on the generation of grammar rules from the ontologies. The rules are generated considering all the elements in the knowledge base, including instances.

3.6.4 PowerAqua (Lopez et al. [117])

Description: PowerAqua is a question answering (QA) system focused on querying multiple ontologies (open domain QA) on the Semantic Web. The input for the system is based on natural language queries. In the first step a linguistic component translates the natural language query into triples (called linguistic triples in [117]). In the second step, a module for Ontology Discovery searches for the possible candidate ontologies likely to contain the information requested in the user query. This module uses WordNet based approximate matching between the user query terms and the terms present in the candidate ontologies. The Semantic Filtering component verifies the validity of the syntactic mappings generated from multiple ontologies in relation to the query terms, generating a set of entity mapping tables connecting query terms to ontology terms. After this mapping process, the entity mapping tables and the linguistic triples are used by the *triple similarity* component to extract a set of ontologies which jointly cover the user query [118]. This module generates as an output a set of triple mapping tables which relates *linguistic triples* to all the equivalent *ontology triples*. These triple patterns are used to query the datasets. The information is later merged and ranked.

The core functionalities for crossing the semantic gap between user queries and datasets which is the focus of this work are concentrated in the PowerMap component. According to Lopez et al. [118], PowerMap is a hybrid matching algorithm comprising terminological and structural schema matching techniques with the assistance of large scale ontological or lexical resources. The mapping process consists of three phases. The first phase has the objective of identifying candidate mappings for the query terms in different ontologies. The matching is based on a string similarity approach, using edit distance and WordNet to lookup synonyms, hypernyms and hyponyms. From the set of ontologies and mappings identified in the first phase, PowerMap excludes terms which are not semantically consistent, exploring the ontology hierarchies to determine the correct sense of the word and using WordNet-based semantic similarity measures to map between query terms and ontology classes. The third phase has the objective of

identifying mappings that better represent the query domain determining the ontologies which better cover the query domain.

Evaluation: The second type of evaluation focuses on the analysis of the capability of the system to answer queries using information provided by multiple datasets defined by multiple vocabularies. The evaluation primarily assesses the ability of the system to map a user query into ontology triples in real time. A dataset of 69 questions was built manually by asking volunteers to build questions based on data available in one or more datasets. Each question is self contained with no references to previous queries. Just precision is measured. The ontologies are based on datasets such as SWETO_DBLP or SWETO (approximately 800.000 entities and 1.600.000 relations). 2GBs of data stored in 130 sesame repositories were collected. In this evaluation the authors report an average precision of 69.5% of the queries with an average execution time per query of 15 seconds.

In a second evaluation, PowerAqua is also evaluated using the *Question Answering over Linked Data* (2011) test collection [119], where precision and recall are collected over a set of 50 natural language queries over DBpedia. The schema size of DBpedia, containing thousands of open domain predicates and millions of instances provides a suitable dataset for the evaluation of schema-agnostic scenario. Queries within the QALD test collection explore complex natural language query patterns.

Critique: The system was one of the first NLI for Semantic Web/Linked Data and defined a basic architecture for QA over Semantic Web/Linked Data. The design of PowerMap, the core mapping mechanism which is the element responsible for matching the query terms to vocabulary terms, relies on WordNet and taxonomical information within the ontologies for computing semantic approximations. In relation to the vocabulary gap, many query-dataset mappings transcend synonymic or taxonomic relations [120, 121], this provides an incomplete solution for the semantic matching problem.

3.6.5 Freya [122]

Description:

Exploring user interaction techniques, FREyA [45], is a question answering system which employs feedback and clarification dialogs to resolve ambiguities and improve the domain lexicon with the help of users. User feedback is used to enrich the semantic matching process, by allowing manual query-vocabulary mappings.

The query processing workflow starts with the syntactic parsing and analysis (using the *C-structures* from the Stanford Parser) (*question analysis* component). The question analysis components also identifies candidate question terms which can be potentially mapped to dataset concepts, using a rules-based approach over the query the C-Structure and POS Tags. The *ontology-based lookup* component maps question terms to the dataset entities. If the system fails to generate a mapping between question and dataset, it returns a dialog to the user through the *consolidation algorithm* component. The consolidation algorithm detects ambiguous concepts (through the disambiguation dialog) or provides alternative mappings based on a relaxation of the query-ontology mapping using neighboring terms (mapping dialog). The mappings are stored by the combination of the question-dataset terms and the surrounding context terms, allowing the improvement of the performance over time. After the mappings are completed, the system detects the answer type and combines the entities identified in the dataset into triples by taking into account the dataset structure (in FREyA there is no strict adherence to the syntactic structure of the question [45]).

Evaluation: FREyA is evaluated using precision and recall under the Question Answering over Linked Data 2011 test collection. Additionally the manual effort involved in the disambiguation dialogs are measured.

Critique: The core contribution behind the FREyA system is the exploration of user interaction/feedback element into for disambiguation and semantic enrichment. The approach is not evaluated with regard to performance and scalability aspects.

3.6.6 ORAKEL [123] & Pythia [43]

Description: In the ORAKEL [123] and Pythia [43] approaches ontologies play a central role in interpreting user questions, i.e. ontological knowledge is used for resolving ambiguities or to interpret semantically light expressions. Both systems depend on an ontology-lexica that make explicit possible linguistic realizations of ontology concepts in a particular language. The meaning of a lexical item is given by reference to an ontology element, i.e. a class, property or individual, providing a separation between the ontological and lexical layer, by using the Lemon vocabulary [124]. ORAKEL relies on *Logical Description Grammars* (LDG) as a syntactic formalism and an extended version of lambda calculus for specifying semantic representations, while Pythia builds on *Lexicalized Tree Adjoining Grammars* [125] (LTAG) as a syntactic formalism and *Dependency-based Underspecified Discourse Representation Structures* [126] for specifying semantic representations. The linguistic representations, in addition to the domain-independent representations, defines the grammar that is used for parsing and interpreting an input

question. The resulting meaning representations are transformed into formal queries, in particular F-Logic and SPARQL queries.

Evaluation: The quality of the results returned by ORAKEL is evaluated using a small geographical knowledge base and a British Telecom (BT) case study. The ontology used to describe the metadata is the PROTON ontology (which consists of 252 classes and 54 relations). Pythia uses the Tang & Mooney test collection to evaluate the quality of the approach.

Critique: ORAKEL and Pythia thus are domain-specific question answering systems, that require an ontology lexicon for the domain that is queried. The main limitation consists in the effort required for building the support lexica, either manually or semi-automatically. And although a grammar-based interpretation process offers high precision, systems relying on domain grammars usually suffer from limited coverage.

3.6.6.1 TBSL [44]

Description: TBSL [44] relies on a parse of the user question to produce a query template. The core rationale behind the approach is that the linguistic structure of a question together with well-defined expressions in the context of QA over structured data (such as *more than* and *the most*) define a domain-independent structure for the query, that then needs to be filled in with domain-specific vocabulary elements. The linguistic analysis of TBSL relies on a parsing and interpretation similar to Pythia. The parsing generates an underspecified semantic representation of the natural language question (without the attachment of specific vocabulary elements) that is then converted into a SPARQL query template. The conversion relies on rules based on lexical categories mappings, where verbs usually correspond to properties, that nouns and adjectives correspond to classes or properties, and that proper nouns correspond to instances.

In order to obtain the vocabulary mappings, TBSL uses an index to look up entities that match with the required class and given natural language expressions. The matching combines string similarity, matching with WordNet synonyms, and on an existing collection of natural language patterns, BOA [127], which collect possible verbalizations for properties based on patterns containing dataset instances that occur in a corpus.

Evaluation: TBSL is evaluated using the Question Answering over Linked Data (QALD 2011) test collection.

Critique: TBSL exploits both natural language and information retrieval techniques and explores corpus-based patterns to support schema-agnosticism. The main limitation resides on the fact that the entity mention-based pattern matching approach

concentrates on a vocabulary enrichment approach highly dependent on the relationship between mentions and associated predicates, not providing a principled approach to address the semantic and commonsense reasoning problem.

3.6.7 Question Answering Systems

The systematic analysis that we employed on the literature review concentrates on natural language mechanisms over structured data. This section briefly introduces existing works on Question Answering systems over unstructured text and spoken Dialogue Systems.

There are three main categories for the construction of Question Answering systems over texts: (i) IR-based QA ; (ii) knowledge-based QA and (iii) hybrid approaches, which combine the two previous approaches.

The first category, QA based on passage retrieval, uses information retrieval models over text passages as part of the pipeline for matching natural language queries to the databases. The typical pipeline consists of a *question analysis component* which detects features such as lexical answer types and which generates a keyword-based query form. The keyword-based query is sent to the a search engine which returns a set of candidate passages which are later processed by an information extraction component. The information extraction component extracts the entities which are candidate answers and ranks them according to their semantic proximity to the lexical answer type and according to other linguistic features present in the query. A list of different passage-based retrieval techniques and the associated QA systems are available in [128, 129].

Knowledge-based QA systems derive logical forms from the textual representation and matches the logical form of the query against a logic-based KB. The question analysis step consists in transforming the natural language question into a logical form query, which is issued over a KB of logical forms extracted from natural language texts. Hybrid QA systems combine both IR and knowledge-based approaches into a single QA pipeline. IBM Watson [130] is an example of a hybrid QA pipeline which uses machine learning techniques to support the combination of complex pipelines for question analysis and answer ranking. Other hybrid systems are: [131] and [132].

3.7 Visual Query Interfaces for Semantic Web/Linked Data Datasets

An important category of query mechanisms allow users to specify queries or progressively filter query results with the help of visual elements in the interface. The approaches in this category focus on addressing the semantic gap problem from the perspective of user interaction, where users can navigate through the data or refine a structured query by providing feedback through elements in a graphic user interface.

3.7.1 Semantic Crystal (Sprenger et al [38])

Description: Semantic Crystal [38] is a domain-independent graphical query interface for querying OWL-based datasets. Users compose queries in Semantic Crystal by building a graphical representation of the SPARQL graph pattern with the help of a visualization of the ontology. The current representation of the query is displayed in a SPARQL dashboard and the user follows the procedure of building the query by clicking the nodes in the ontology and selecting possible elements in the menu. Users can assemble the query in the ontology visualization component or in the SPARQL dashboard. The approach is based on the TORC approach [133], where the interface supports four fundamental elements: *token* (classes), *output* (output SPARQL values bound to variables), *restriction* (values in datatype properties and SPARQL “FILTER” statements) and *connection* (object properties). Before users can query, the specified ontology need to be loaded into Semantic Crystal. The T-Box of the ontology is converted into a GraphML model, an XML used for the visualization component.

Evaluation: Semantic Crystal is evaluated in [38] through a comparative usability study against three other systems (NLP-Reduce, Querix and Ginseng). The collected metrics included average time per query creation/reformulation, average number of queries, average success rate, average failure rate, average number of successful queries per minute, average number of failed queries per minute. The data collected in the usability study was analyzed quantitatively and qualitatively. Users were also evaluated in terms of a system usability score (SUS), created from a questionnaire. The quantitative part included ANOVA, T-test and Mixed Linear Regression Models.

Critique: Semantic Crystal is a graphical SPARQL query composer. The tool demands from users less formal effort compared to the alternative scenario of directly writing a SPARQL query. The approach, however, does not abstract users from the conceptual model behind the dataset. The approach supports a similar level of query expressivity to SPARQL. Since it is a graphical SPARQL builder, the idea of an approximate query is

not supported by the system. The suitability of the approach for querying large datasets is not verified.

3.7.2 QUICK (Zenz et al. [134])

Description: QUICK [134] is an incremental query construction tool targeted towards RDF. The core process behind QUICK is the transformation of a keyword based query into a final graph pattern (semantic query) using incremental user feedback through a visual interface. For a given dataset, QUICK starts generating the set of all possible query templates based on the dataset schema. In the second part, the keywords are bound to the schema elements and then a set of possible query interpretations are displayed on the interface for user disambiguation. For each step, a set of query guides, possible query sub-patterns that should be selected by users, is generated. The final query graph pattern is built incrementally by user interaction. The system does not apply semantic approximation in the query formulation process.

Evaluation: QUICK was evaluated using two datasets: the Internet Movie Database, IMDb (5 classes, 10 properties, 10 million instances and 40 million triples) and Lyrics (3 classes, 6 properties, 200 thousand instances and 750 thousand triples). The query log of the AOL search engine was used: 3000 queries associated with IMDb and 3000 queries associated with Lyrics were selected as the query dataset. Since most of the queries were not expressive enough to evaluate the scenario of the semantic queries, Zenz et al. manually curated 100 queries for IMDb and 75 queries for Lyrics, varying from 2 to 5 keywords. Keyword queries were defined as inputs and two metrics were measured, computing the associated selection evaluation cost and the number of selections users had to make to build the final query. The second set of experiments was focused on the measurement of the efficiency of the query generation process. The response time for each interaction of the query construction process was recorded. In addition, the quality of the generated query guides was evaluated.

Critique: The evaluation of the work is focused on querying datasets with small schema sizes. In a schema-agnostic scenario, where users query open domain, large-schema datasets, the proposed approach is likely to present scalability problems: the space of query guides can potentially become too large for users selecting query sub-patterns and for the computation of query suggestions.

3.8 Analysis & Gap Identification

Table 3.8 shows a summary of the comparative analysis of existing approaches with regard to the set of requirements for schema-agnostic queries. Approaches under different categories have covered different requirements dimensions. Table 3.8 summarizes the techniques used in the analyzed set of query and search mechanisms, also classifying each approach with regard to query expressivity and performance/scalability.

Category	Approach	Requirements		High query expressivity	High scalability	Interactive & search & Low query-execution time	Accurate & comprehensive semantic matching	Low setup & maintainability effort	High usability & Low query construction time
		Level of schema-agnosticism							
Entity Search	Tummarello et al. (Sindice)	-		+-	++	++	-	++	+-
	Harth et al. (SWSE)	-		+-	++	++	-	++	+-
	Kiefer et al. (iSPARQL)	-		++	NE	NE	-	NA	-
	Dong & Halevy	+-		+-	+	++	+	++	++
	Zhang et al. (Semplore)	+-		+-	++	++	-	++	+-
Approximate Queries	Zhou et al. (SPARK)	+		+-	NE	NE	+	++	++
	Stuckenschmidt & van Harmelen	-		++	NE	NE	NE	++	-
	Hurtado et al.	-		++	NE	NE	NE	NA	-
	Oren et al.	-		++	+	NE	-	++	-
	Damijanicovic et al. (Freya)	+		+	NE	+	+	-	+
Natural Language Interfaces	Cimiano et al. (ORAKEL) & Unger et al. (Pythia)	+-		+	NE	NE	+-	-	+
	Unger et al. (TBSL)	+		+	+	+	+	+	++
	Lopez et al. (PowerAqua)	+		+	+	+	+	+	++
	Kochut & Janik (SPARQLer)	-		path queries	NA	NA	-	++	-
	Kaufmann et al. (NLP-Reduce)	+-		+-	-	NE	-	++	++
Visual Query Interfaces	Kaufmann et al. (Querix)	+		+	NE	NE	+	++	++
	Bernstein et al. (Ginseng)	+-		+	+	NA	-	++	+-
	Sprenger et al. 2007 (Semantic Crystal)	-		+	-	NA	-	-	-
	Zenz et al. (QUICK)	+-		+	-	NE	NA	++	-

Legend:

++ : requirement dimension is well covered.

+ : requirement dimension is partially covered with positive results.

+- : there is an attempt to address requirement dimension but the solution is not effective.

- : the requirement dimension is poorly covered.

- : the requirement dimension is very poorly covered.

NA : the requirement dimensions is not addressed or focused on the research.
NE : a dimension dependent on an evaluation is either poorly or not evaluated.

Categories	Approach	Feature Semantic Approximation	Supporting Knowledge Bases/Linguistic Resource	Reasoning	Disambiguation	Ranking	Performance / Scalability mechanisms	Query Type	Schema Size
Entity Search	Tummarello et al. (Sindice)	None	None	None	None	Modified TF/IDF + Link-based	Inverted Index	Keyword/Star-shaped	Large/Multiple Datasets
	Harth et al. (SWSE)	None	None	OWL Authoritative	None	Modified TF/IDF + Link-based	Inverted Index	Keyword/SPARQL	Large/Multiple Datasets
	Kiefer et al. (IS-PARQL)	Levehnstein, Jaccard and TF/IDF (Synonyms)	None	None	None	None	NF	SPARQL	Small/Single Dataset
	Dong & Halevy	Query/Dataset Term Expansion	WordNet	None	None	Yes	Inverted Index	Keyword with structure information	NA
	Wang et al. (Semplore)	Dataset Term Expansion (Taxonomical Enrichment)	None	OWL	Facet-based	Yes	Inverted Index (Position Index)	Keyword with structural information (single-atom queries, path queries, star-shaped queries, entity queries and tree-shaped queries)	Large/Enumerated List of Datasets
	Zhou et al. (SPARK)	Dataset Term Expansion/Edit distance, Substring matching	WordNet	None	None	Yes		Keyword-based	Small/Single Dataset
Approx. Queries	Stuckenschmidt & van Harmelen	Logical/Query-Data Structural Relaxation	None	Taxonomical	None	None	None	Structured	Not evaluated
	Hurtado et al.	Data Constraint Relaxation (Taxonomical)	None	Taxonomical	None	None	None	SPARQL	Not Evaluated
	Oren et al.	Constraint-based (evolutionary)	None	NF	None	None	Bloom filters	Structured	Small/Enumerated List of Datasets
Natural Language Interfaces	Damjanovic et al. (Freya)	Query/Dataset Term Expansion, Manual lexicon enrichment	WordNet	None	Disambiguation dialog	Yes	NA	Full Natural Language	Large/Single Dataset
	Cimiano et al. (ORAKEL) & Unger et al. (Pythia)		Manually created Lexica	Ontological	None	None	No	Full Natural Language	ORAKEL (Small) Pythia (Large)/Single Dataset
	Unger et al. (TBSL)	Corpus pattern mining	WordNet, Corpus-based			Yes	Yes	Full Natural Language	Large/Single Dataset
	Lopez et al. (PowerAqua)	WordNet-based (hyponym, synonym) Semantic/String Similarity, based on taxonomical relations in the data	WordNet	Ontological	None	Based on the similarity Scores	NA	Full Natural Language	Large/Single Dataset
	Kochut & Janik (SPARQLer)	Regex based	None	None	None	None	No	Structured Path Queries	Small/Single Dataset
	Kaufmann et al. (NLP-Reduce)	String similarity, edit distance	None	None	None	None	No		Small/Single Dataset
	Kaufmann et al. (Querix)	Query/Dataset Term Expansion	WordNet	None	Yes	None	No	Full Natural Language	Small/Single Dataset
	Bernstein et al. (Ginseng)	None	None	None	During the manual selection	None	No	Controlled Natural language	Small/Single Dataset
Visual Query Interfaces	Sprenger et al. 2007 (Semantic Crystal)	Manual Selection	None	None	During the manual selection		No	Visual structured queries (SPARQL)	Small/Single Dataset
	Zenz et al. (QUICK)	Manual Selection	None	None	Yes	NF	No	Visual structured queries (SPARQL)	Small/Enumerated List of Datasets

The core characteristics of existing approaches can be summarized as follows:

Natural Language Interfaces for Semantic Web/Linked Data Datasets: Natural language interfaces (NLI) query mechanisms have concentrated on approaches which provide higher levels of schema-agnosticism, better exploiting semantic techniques to address schema-agnosticism. Most of the existing natural language query systems implement some type of WordNet-based semantic approximation. Strategies include query/dataset term enrichment (hypernym, hyponym, synonym) or the computation of WordNet-based similarity measures. WordNet-based approaches are usually complemented by the use of taxonomic information present in the dataset.

The first limitation of these approaches involves relying purely on WordNet to cope with the lexical/semantic approximation process. Approaches relying in WordNet are limited in: (i) domain and language transportability; (ii) restricted to semantic approximations techniques which strongly rely on taxonomic/synonymic relations (semantic similarity); (iii) semantic similarity measures limited in addressing the computation of similarity between terms crossing part-of-speech boundaries or containing multi-word expressions and (iv) ability to capture uncommon and new terms or term senses.

Some works [45, 111] have explored the use of user interaction elements for supporting the resolution of ambiguities in the interpretation of the query.

Another limitation of NLI approaches is the lack of a principled mechanism for coordinating both the syntactic-structural and conceptual semantic approximations. While some approaches provide a crisp semantic interpretation process based on well-defined grammars, constraining the dataset structure to be isomorphic to the query syntactic structure [43, 123], other approaches provide a syntactically loose interpretation of the query [10].

The third limitation of some NLI query approaches is the lack of explicit mechanisms and the evaluation of the temporal performance and scalability aspects. While approaches such as [44] employ explicit indexing techniques, many of the approaches do not address this concern.

With regard to the evaluation, test collections such as the Question Answering over Linked Data [135] have provided a proper evaluation benchmark for schema-agnostic queries in the context of NLI, using a large-schema and large-size dataset (DBpedia) and an expressive query set, mapping to different structured query patterns. However, many of existing works are evaluated in the context of a low schema-size dataset (Tang & Mooney). QALD is emerging as a community adopted test collection.

Entity Search: Vector Space Models for Semantic Web/Linked Data Datasets

Entity search approaches have focused on transporting techniques used in information retrieval for searching over Semantic Web/Linked Data. The use of vector space models and the associated data structures (*inverted indexes*) is used to support query execution and indexing temporal performance and scalability. The evaluations concentrated on the performance aspects of the approaches, and existing models have achieved high performance and scalability levels.

Entity search approaches have also started to explore the role of user interaction for disambiguation purposes, in most of the cases using facets.

Most approaches are based on the application of vector search models over Semantic Web/Linked Data datasets and are oriented towards keyword-based queries, not providing schema-agnostic queries with higher expressivity. For this category, higher query expressivity is usually achieved by introducing explicit mentions to elements of the datasets' conceptual model in a semi-structured query format (e.g. star-shaped queries), where schema-agnosticism is traded for query expressivity. Approaches such as [103] started to explore the creation of index structures which can keep the structural information of datasets, while enriching their terms with taxonomic or lexical information.

Existing test collections lack the evaluation of queries with more expressivity, which can support an evaluation of schema-agnostic queries. Most of the approaches put a strong emphasis on the evaluation of performance and scalability aspects.

Approximate Queries for Semantic Web/Linked Data Datasets

Existing approaches under the approximate queries category concentrate on the employment of the relaxation of structural query constraints and on exploiting the semantic information on the dataset to support a taxonomic-based semantic relaxation. Other approaches have focused on providing constructs which introduce approximation operators as a primitive in the SPARQL query. The approximation operators consists mostly of string, taxonomic and structural similarity. The set of analyzed works have employed their techniques with different motivational scenarios, ranging from the discovery of entity associations to semantic relaxation.

Form the evaluation perspective, works range from purely theoretical contributions to empirically supported works. To the extent of my knowledge, there is no commonly adopted test collection for this category.

Visual Query Interfaces for Semantic Web/Linked Data Datasets

Visual query interfaces have concentrated on allowing users to build structured queries by exploring graphical user interface elements. In this case, users are not abstracted from the schema, but the visual interface allows users to do a visual exploration in a

constrained subset of the data. There is empirical evidence that casual users prefer a schema-agnostic NLI approach [116] to the visual interfaces due to the effort necessary to build a structured query using a purely visual editor. There is no empirical evidence that visual query editors can scale to large-schema datasets.

In summary the *main gaps* identified in the literature review with regard to schema-agnostic queries were:

Need for more comprehensive query-dataset semantic matching strategies:

Existing semantic matching approaches currently strongly rely on WordNet and on explicit taxonomical relations in the data to support lexical/semantic approximation in the dataset. As is covered in Chapter 2, the semantic gap between query and dataset transcends the semantic approximations and inferences which can be supported by these two mechanisms (for a systematic analysis on the query-dataset semantic gap, the reader is referred to [120]). Moreover, the scale of the semantic and commonsense knowledge necessary to achieve a generic solution for the query-dataset semantic gap, transcends the scale of WordNet. Aiming for a solution which can provide a more comprehensive semantic matching, without the constraints of manually logically encoding large semantic and commonsense knowledge bases is one major demand for schema-agnostic query mechanisms.

Scalable query-dataset semantic matching: Providing a better communication between information retrieval techniques and semantic matching approaches. Integration of temporal performance and scalability mechanisms developed in the context of information retrieval into the semantic matching approaches. Creation of indexing strategies with embedded semantic matching capabilities.

Principled definition of the query-dataset semantic matching phenomena

(and which dimensions are covered by each approach): The scientific discourse on natural language interfaces over Semantic Web/Linked Data is oriented towards a coarse grained analysis of the full spectrum of the different semantic matching phenomena involved between matching the schema-agnostic query to a dataset. Currently, there is no fine-grained understanding on the strengths of particular approaches with regard to different semantic phenomena involved in the matching of query and dataset, where the quality of an approach is evaluated by the aggregate measures such as f-measure.

Evaluation of query-dataset over large-schema datasets: Despite the emergence of test collections such as QALD in the context of NLIs, many different approaches are evaluated in the context of small-schema datasets (e.g. Tang & Mooney). The validation of schema-agnostic query approaches should be performed in large-schema datasets,

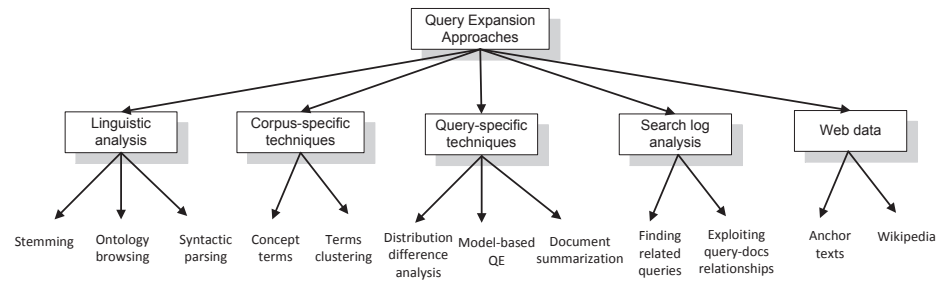


FIGURE 3.1: A taxonomy of approaches to AQE.

where the complexity of the semantic phenomena (ambiguity, synonymy, vagueness) in real world scenarios can be replicated.

The identified gaps were used to position and to maximize the impact the proposed model.

3.8.1 Automatic Query Expansion Strategies

3.8.1.1 Approaches for Automatic Query Expansion

In the previous section we analysed the application of different semantic matching techniques in the context of searching and querying structured data. Many of these techniques were introduced in the Information Retrieval community, under the Automatic Query Expansion (AQE) field. This section describes the main reference literature in the automatic query expansion (AQE) area applied over unstructured text documents.

Carpineto & Romano [11] provide a comprehensive survey on existing automatic query expansion (AQE) techniques. According to Carpineto & Romano, AQE techniques can be classified into five main groups according to the paradigm used for finding the expansion features: *linguistic methods*, *corpus-specific statistical approaches*, *query-specific statistical approaches*, *search log analysis*, and *Web data*. These groups are further specialized into a taxonomic structure depicted in Figure 3.1.

These categories are briefly described in the following sections.

3.8.1.2 Linguistic Methods

Consists of approaches which use properties such as morphological, lexical, syntactic and semantic word relationships to expand or reformulate query terms []. Linguistic analysis models may vary from simple stemming/lemmatization models to approaches

which use external knowledge bases (dictionaries, thesauri or ontologies). Many recent AQE approaches are based on WordNet, where synonyms and taxonomic relations are used in the query expansion process. Word sense disambiguation is a critical part of WordNet-based AQE [11]. *Ontology browsing* techniques, such as the model developed in Navigli and Velardi 2003 [136], use manually constructed conceptual models for AQE, where ontology navigation or deductive reasoning over an ontology is used to support AQE. *Syntactic analysis* techniques consists in using syntactic information (such as dependency or C-Structures) to support the query expansion process Sun et al. 2006 [12].

3.8.1.3 Corpus-specific statistical approaches

Consists of approaches which use the statistics of co-occurrence patterns in textual corpora to establish term correlations, using it in AQE. These correlations are usually established using the target document collection as a corpus, defining correlations at the document level or, in order to better handle topic drift, in more restricted contexts settings such as paragraphs, sentences, or small term neighborhoods. Examples of corpus-specific approaches can be found in (Qiu and Frei 1993 [13], Bast et al. 2007 [14], Crouch and Yang 1992 [15], Schuetze and Pedersen 1997 [16], in Gauch et al. 1999 [17], Hu et al. 2006 [18], Park and Ramamohanarao 2007 [19], and Milne et al. 2007 [20], which make use of *context vectors*, *mutual information*, *latent semantic indexing*, and *interlinked Wikipedia articles* [11].

3.8.1.4 Query-specific statistical approaches

Query-specific techniques take advantage of the local context provided by the query [11]. Most query-specific techniques are applied over top-ranked documents.

Model-based AQE techniques consists on the construction of a statistical language model for the query, specifying a probability distribution over terms, in which terms with high probabilities are expanded. The two main representatives are the *mixture model* (Zhai and Lafferty 2001 [137]) and the *relevance model* (Lavrenko and Croft 2001 [138]) and both make use of the top retrieved documents [11]. In mixture models, a query topic model is built from the top-ranked documents by extracting the part that is most distinct from the whole document collection.

Document summarization approaches consist on several methods for finding more compact and informative document representations, such as *passage extraction* (Xu and Croft 1996 [23]) and *text summarization* (Lam-Adesina & Jones 2001 [21]). In Chang

et al. 2006 [22] apud [11], the document summaries are clustered in order to reduce the set of orthogonal features describing each document.

3.8.1.5 Search log analysis

Search log analysis approaches mine query associations that have been implicitly suggested by user query patterns. Search logs provide an implicit relevance feedback. As Carpineto & Romano summarizes [11]: “On the other hand, implicit measures are generally thought to be only relatively accurate (see Joachims et al. 2007 [139] for an assessment of the reliability of this assumption) and their effectiveness may not be equally good for all types of users and search tasks (White et al. 2005 [140]). Other problems with their use for AQE are caused by noise, incompleteness, sparseness, and the volatility of Web pages and query (Xue et al. 2004 [141]). Also, the availability of large-scale search logs is an issue.”

3.8.1.6 Web data

A common Web data source for AQE are *anchor texts*. There is an intrinsic similarity between anchor texts and real user search queries as most anchor texts are succinct descriptions of the destination page [11]. Kraft and Zien [24] analyses several ranking criteria for anchor texts which are data-specific (e.g. such as the number of occurrences of an anchor text) [11]. Arguello et al. 2008 [142] apud [11] proposes a method based on Wikipedia documents and anchor texts.

3.8.1.7 Applications of Automatic Query Expansion techniques for structured data

Most of the semantic matching approaches for searching and querying structured data concentrate on *linguistic methods* and query-specific statistical approaches. This work concentrates on the application of *corpus-specific statistical approaches* using *Web data* for supporting the query-data semantic matching for queries over structured data. Differently from traditional IR *corpus-specific statistical approaches* which employ a bag-of-words model, not taking into account the compositional structure of the sentences in the text, this work explores word vector models for representing the semantics of dataset entities, supported by the compositional information that can be derived by the structure of the data. Additionally, most works in *corpus-specific statistical approaches* uses the statistical term correlations present in the target document collection. In this

work the scale of available web data external to the target dataset is used as a semantic resource to create the distributional word vectors.

3.9 Chapter Summary

This chapter analysed the state-of-the-art for querying and searching structured data with regard to schema-agnosticism. Different categories of approaches including Natural Language Interfaces, Approximate Query Mechanisms and Entity Search over structured data were analysed relative to the set of core requirements for schema-agnostic queries. The main mechanisms used for semantic matching were identified. This analysis supported the identification of the main gaps in the literature which include: (i) *need for more comprehensive query-dataset semantic matching strategies*; (ii) *investigation and evaluation of scalable techniques for query-dataset semantic matching*; (iii) *principled definition of the dimensions involved in the query-dataset semantic matching phenomena* and (iv) *evaluation of query-dataset over large-schema datasets*. The contribution of this thesis concentrates on the proposal of a schema-agnostic query approach which addresses these gaps fully or partially. Associated publications to this chapter are [143, 144].

Chapter 4

Towards a New Semantic Model for Databases

“An educated mind is distinguished by the fact that it is content with that degree of accuracy which the nature of things permits, and by the fact that it does not seek exactness where only approximation is possible.”

Aristotle

4.1 Introduction

Current approaches for querying databases are dependent on a perfect syntactical and lexical matching, where the semantics of the query-database match is simplified under a perfect symbolic and syntactic contract. The support for schema-agnostic queries is dependent on a *principled semantic model* to address the *query-database semantic matching*, revisiting the semantic and semiotic assumptions behind database querying. This chapter focuses on the analysis of the semiotic and semantic assumptions behind existing database query models and on the investigation of the general requirements and characteristics of a semantic model that can support schema-agnostic queries under data environments with high *schema size, complexity, dynamicity and decentralisation* (SCoDD).

This chapter starts by analyzing the semiotic and semantic assumptions behind databases today (Section 4.2.2). The set of requirements for a semantic model to support schema-agnostic queries under the SCoDD conditions are defined in Section 4.3.3. In order to

support a robust semantic matching mechanism for schema-agnostic queries, different perspectives on semantics are described, analyzed and compared against the set of requirements (Section 4.4). The structuralist perspective of semantics is discussed in the context of distributional semantics models, which is described in Section 4.7.

Based on the characteristics and by composing complementary aspects of different semantic models, a new semantic model for databases is proposed. The proposed model integrates the structuralist/distributional perspective on semantics to the logical perspective, in order to provide a semantic model which supports schema-agnostic queries. This chapter paves the way to the formal construction of the hybrid distributional-relational model for schema-agnostic queries ($\tau - Space$), which is defined in Chapter 6.

4.2 A Semiotic Model for Databases

4.2.1 Semiotics

Humans communicate meanings through the generation and interpretation of *signs*. A sign can be defined as a symbol referring to or standing for something other than itself, or as anything that in a certain way or aspect, represents something to someone [145]. The idea of semiotics was introduced in the nineteenth century by Charles Peirce in his work *Logic as Semiotics: The Theory of Signs*. Semiotics considers how signs are created and how they are used to store and to transmit information. According to [146]: *semiotics* concentrates on the study of the basic aspects of cognition and communication.

Saussure [147] defines a dyadic model of the sign, where a sign is composed of: (i) a *'signifier'*, the form which the sign takes and (ii) the *'signified'*, the *concept* it represents. The sign is the whole that results from the association of the signifier with the signified [147]. The relationship between the signifier and the signified is referred to as *'signification'*. The Saussurean view of meaning is not to be identified with the referent (object in itself), but with a mental conceptualization of it. The term symbol is used to refer to the linguistic sign. Figure 4.1 depicts the *semiotic triangle*, that shows the relationship between the signifier, the signified and the referent object. According to [148] *apud* [147]: *'Symbols are not proxy for their objects but are vehicles for the conception of objects... In talking about things we have conceptions of them, not the things themselves; and it is the conceptions, not the things, that symbols directly mean'*.

Andersen [150] provides a model for computer semiotics which views computer systems as sign-vehicles, whose main function is to be perceived and interpreted by some group

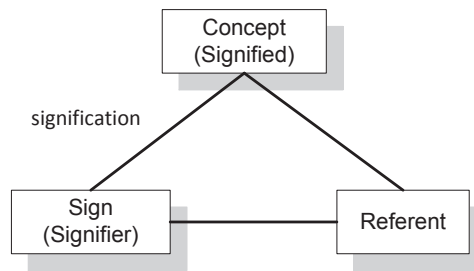


FIGURE 4.1: Semiotic triangle (Loebner, 2014 [149]).

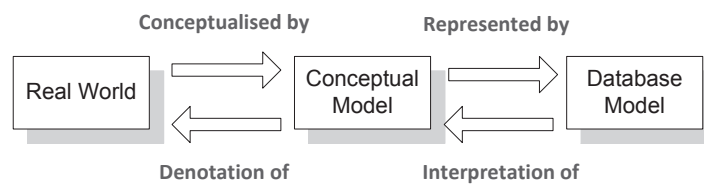


FIGURE 4.2: Projection of the semiotic triangle to database (George, 2005 and Sheth & Larson, 1990 [87, 88]).

of users. In this context, semiotics has nothing to say about data in itself, only in its capacity of being interpreted and used as a source of knowledge or guide for action. Computer systems are symbolic machines constructed and controlled by means of signs [150].

4.2.2 Semiotics for Databases

The database is an information artefact which is used to persist data under a *structured representation*. From a semiotics perspective, the way data is represented in databases induces a specific semiotic function to databases. The *structured data representation format*, which uses a data model, facilitates the *retrieval*, *processing* and *analysis* of the information stored in databases by both human and different software systems.

Databases are used to communicate a precise understanding of a domain of discourse between different humans, systems or between humans and systems in the context of a specific task. In databases the communication function is mixed with the semantic function: the semantic representation for the data serves both as a communication and as a semantic representation function, which is used for *retrieval*, *processing* and *analysis*.

Figure 4.2 depicts the semiotic functions of different database elements as a projection of the semiotic triangle for databases [87].

From the communication perspective, two main communication scenarios for databases can be distinguished:

- **Closed communication:** where the symbolic system of the database is known a priori by the user, which interprets it in an unambiguous way. In this case, typically there is a close contextual proximity between the database symbols and the data consumer (e.g. a software developer developing an application). The context in which the data is consumed is well-defined and it is typically the same context in which the data was created.
- **Open communication:** where the symbolic system of the database is unknown by the user. In this scenario there is a clear separation between data consumer and the database symbols, which were typically produced by a third party, under a different context from the data consumer (e.g. a data journalist reusing government data). In this scenario, the data consumption task can be held in different contexts.

In many data consumption scenarios the user is located in the middle of the closed/open communication spectrum. In this case the user may have a partial knowledge of the symbolic system expressed in the data, or the terms used in the symbolic system may have changed over time. This work concentrates on the process of coping with the incompleteness associated with the symbolic part of the dataset that is not known a priori by the user.

Databases are typically created under a closed communication scenario and can be reused by third parties under both an open and closed communication scenario. The *open communication* scenario maps to the *SCoDD conditions* and it is the communication scenario which is the main target of this work.

The shift from a scenario where communication is performed in a *single context* to a scenario where data is generated and consumed in *multiple contexts*, in addition to the increase of the size of database schemas and in the number of data sources, drastically changes the assumptions on how users communicate with databases, which today is heavily grounded on structured queries.

The communication process associated with database querying starts with the definition of an *information need* or *query intent* which is based on a *task* under a specific *context*. The information need is expressed according to a representation under a human cognitive conceptual model. Different conceptual models entail different conceptualizations of the reality. The mental representation of the information need can be directly translated into a natural language query following the syntax of a specific language. In an open

communication scenario, the user needs to translate his mental representation into a structured query under the database lexicon, which depends on a previous learning of a database structured query language syntax, and of the understanding of the database lexicon and structure in which the conceptual model is expressed. This last step is done by the manual exploration of the database schema, using a representation of the schema and its associated natural language descriptors. The user interprets the database lexicon, aligning them with his cognitive model.

The elements involved in the human database communication can be organised into a high-level abstract model, describing the relationship between different elements which impact in the communication process. The model, depicted in Figure 4.3 consists of the following elements:

- **Representation Model:** Consists of the combination of the **conceptual model** (database schema), **data model** (syntactic dimension) and the **database physical model**. From a semiotic perspective, the physical model supports the expression of the conceptual and the data model, but users do not interact with it (George, 2005 and Sheth & Larson, 1990 [87, 88]).
- **Computational Model:** Consists of the data transformation operations supported by the database. In most databases it consists of *solution modifiers* (e.g. conditional and aggregation operators).
- **Query Language Syntax:** Consists in the syntax of the query language which is used to query the database. The syntax is dependent on the data model. In most cases the query language exposes the basic syntax of the data model, extending it with the solution modifiers.
- **Database Lexicon:** Consists in the set of terms which are in the conceptual model and the terms which express the database operations.
- **Database Structure:** Consists in the syntactic relations between terms in the lexicon expressed under the data model syntactic constraints.
- **Natural Language Syntax:** Consists in the syntax of the natural language used by the data consumer.
- **Data Consumer Lexicon:** Consists of the lexicon of the data consumer.
- **Task:** The task which must be addressed by the data consumer with the information in the database.
- **Query Intent:** The query intent is the information need which is generated by the task.

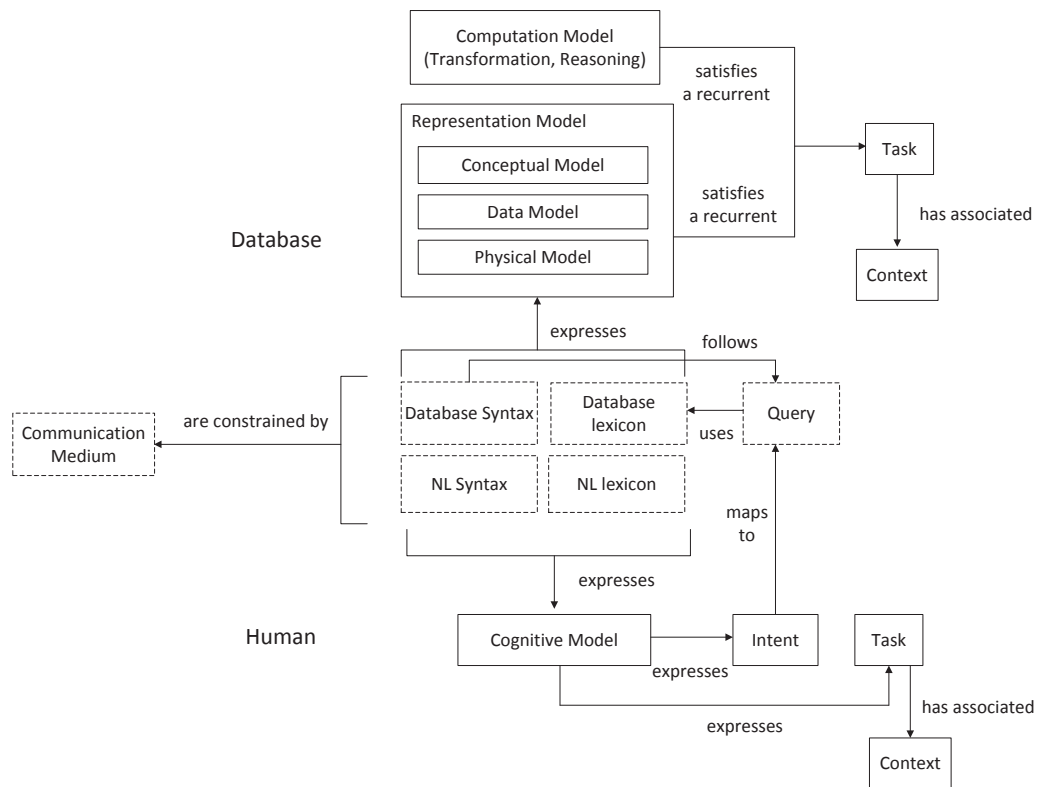


FIGURE 4.3: Elements involved in the human-database communication.

- **Query:** The materialization of a query intent under a natural language or under the syntax of a structured query language.
- **Context:** The domains of discourse which are associated with the *tasks*, associated to the data and the query.

Figure 4.3 depicts the relationship between the elements of the semiotic model in the context of the human-database communication. In this model it is assumed that query and database are under the same language.

The cost of querying a database is proportional to the cost of aligning the user lexicon to the database lexicon, satisfying the structural constraints between the database lexical items, added to the constant cost of expressing the query under a structured query language.

The process of mapping the human to the database lexicon is dependent on the semantic phenomena associated with human language (*lexical and structural ambiguity, synonymy, vagueness*) also taking into account the difference of contexts between user and database (Figure 4.4). In the open communication scenario, the impact of the semantic phenomena in the mapping process between database and user lexicon grows due to the increase of schema size and contextual differences.

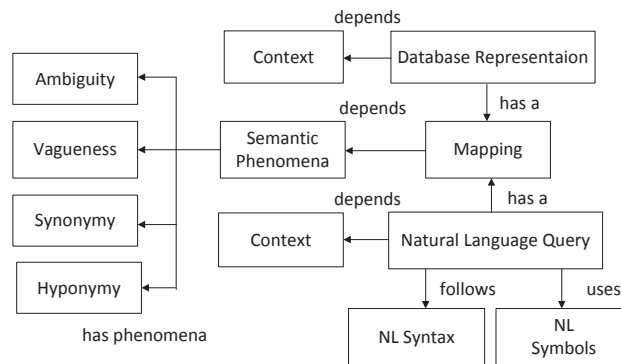


FIGURE 4.4: Core elements and concepts involved in the human-database communication. Focus on the elements related to the natural language aspects of the user-database communication.

In this work the cost associated with the user-database lexical alignment is not quantitatively and empirically measured. The dependencies expressed in the semantic entropy measures described in Chapter 5 can be used as estimators for a qualitative analysis of this effort.

In the next sections we analyze the semiotic and semantic assumptions behind three dimensions in the data representation dimension, with a particular focus on the assumptions behind Semantic Web/Linked Datasets (the reference data model): *symbolic grounding*, *data model* and *conceptual model*.

4.2.3 Symbolic Grounding in Databases

In the closed communication scenario, the database lexicon which materializes the conceptual model is represented by unique identifiers for the concepts in the conceptual model. Under the perspective of the Semantic Web/Linked Data Web (the data model which is used in the discussion of this work), the database elements can have associated URIs, which can make them visible and referenceable outside the original database context.

The namespace which defines the URI is the authoritative source of its meaning [99], and defines an identifier for the context in which the URI is defined. The unicity of its meaning is built-in on the naming mechanism of the Web (the Domain Name Systems, DNS¹) and provides a Web-scale mechanism to support unambiguous referencing to database elements. The set of URIs for a database defines the symbol space that users need to map their internal conceptualizations into, in order to query the data.

¹http://en.wikipedia.org/wiki/Domain_name_system

Considering the *open communication scenario*, the user has six possible information sources to interpret the meaning of a third-party URI (mapping to his conceptualization space):

- *Direct descriptors:*
 - The natural language content embedded in the URI string.
 - The natural language descriptors associated with the URI.
 - The metadata associated with the URI (e.g. provenance).

- *Associated descriptors:*
 - Terminology-level elements associated with the URI and their direct and associated descriptors.
 - Instance-level elements associated with the URI and their direct and associated descriptors.
 - Software constructs and textual elements referencing the URI.

Users querying third-party data are dependent on the six interpretation sources in order to interpret the meaning of a URI. In this case the interpretation of the database schema is *mediated by interpretation sources which are dependent on the natural language descriptors* and their associations. By being dependent on the interpretation of natural language descriptors outside its original creation context, these sources are subject to the semantic phenomena associated to natural language (ambiguity, vagueness, hyponymy and synonymy) in the interpretation process. This is supported by the empirical evidence that concepts of a given vocabulary are used outside its strictest intended sense [151].

For the reasons above, the URIs as a semantic and semiotic system have the following limitations:

- *Growth of the symbolic space \mathcal{E} associated cost of reuse:* URIs provide a unique identifier for the context in which a concept is used. Humans querying datasets, building systems and mapping different conceptual models need to go through the process of mapping their internal conceptualization to the conceptual model URI, a process which is mediated by the interpretation of natural language descriptors. For the open communication querying scenario, URIs do not provide a semantic mechanism.

The strength of URIs as a semiotic mechanism lies in the following elements:

- *Built-in interpretation scoping mechanism:* The namespace (prefix) associated with a URI can be used to indicate the scope for the unambiguous interpretation of that symbol.
- *Universal (Web-wide) unique referencing mechanism:* Which supports its referenceability at Web-scale, which is supported by the DNS infrastructure.
- *Structured description:* Under the Linked Data context, de-referenceable URIs are able to provide a structured description under the RDF(S) standard, which can be used to define a semantic approximation mechanism between different URIs (e.g. taxonomic reasoning and identity links).

4.2.4 Data Model

Many data models for databases are grounded in a relational/predicate-type representation which can be mapped to first-order logic [66]. The *predicate-argument* structure which is the common ground across different data models can be associated to how information is cognitively organised and processed in the human brain, which defines the basis of language and logic. Research in the cognitive sciences showed that there is evidence that there is a neural correlate for the predicate-argument structure [152].

From a semiotic perspective, data models provide the structure in which conceptual models can be expressed. In the context of databases, data models are structures used for representing concepts, objects, their attributes and relationships in a domain of discourse in such a way that operations such as selection, filtering, comparison, aggregation can be facilitated.

Data models constructs provide a representation framework which supports:

1. **Concept definition:** Where words can be composed to form a database primitive concept associated with a domain. A primitive symbol maps to the individuation of primitive concepts in the database (e.g. ‘previous employees’ to describe a database relation).
2. **Distinction between different entity categories:** Use of distinct data model types to represent different categories of concepts. The most basic level of differentiation is between predicate and constants (for example in the datalog model), where predicate-type entities describe classes and relations and constant-type entities are named entities and numerical values. Other data models, such as RDF(S)

make additional distinctions (class, instance, property, value) or the relational model (relation, attribute, value, key, etc). A tuple is the atomic statement in the database, corresponding to the instantiation of a set of predicates.

3. **Syntax:** Consists in the core compositional/syntactic pattern of the data model, describing how elements from different categories can be combined, i.e., it describes how the tuple is formed.
4. **Collection:** Describes a collection of tuples. In a relational model a table is a collection, while in RDF(S) the collection is defined by a graph.

The data model ultimately impacts on the ability to address a particular type of task. The relational data model and its core associated visualization structure (the tabular structure) better expresses domains which are semantically homogeneous, expressing a less variable and dynamic set of attributes [5]. The table as a communication device emphasizes the categorization of individuals (relation name) and a rigid associated set of attributes. This data model facilitates operations such as ordering, filtering, aggregation and comparative visualization over elements containing the same set of attributes.

The RDF(S) data model and graph structure facilitates the construction of databases which are semantically more heterogeneous (i.e. which can express high variability in attribute composition for the instances in the database). The support for complex schemas also reinforces the integration of different databases on the Web (Linked Data). The increase in complexity shifts the core structure from relations in the relational space (a set of predicates associated with an entity type) to instances in the Linked Data space.

The RDF(S) data model also facilitates a more expressive semantic representation of the data by allowing terminology, instance-level and metadata under the same graph representation. In particular, RDF(S) facilitates the description of terminology-level relationships, bringing an additional level of structured description to terminology-level elements (e.g. taxonomical description), enhancing the interpretability of these elements for both human and computer agents.

Data models based on predicate-argument structures reflect an isomorphism with the syntax of human language. There is a typical correspondence between different categories of database elements and lexical categories. Table 4.2.4 provides a partial correspondence between categories of different data models and some of their frequently corresponding lexical categories [153].

RDF(S)- EAV/CR	Logical	Relational	Natural Language
Instance	Constant	Value	NNP+
Value	Constant	Value	CD+
Class	Unary predicate	Entity, Attribute	RB+—JJ+ NN(S)+ IN NNP+
Property	Binary predicate	Entity, Attribute, Relation	BE VB IN, BE VB NN+

TABLE 4.1: Correspondence between RDF, Logics, Relational and Part-of-Speech (lexical categories) Patterns.

4.2.5 Conceptual Model

The *conceptual model* provides a description of the concepts and relationships in a domain of discourse. The description of a conceptual model can be formalized using different notations and represented using different data models. Under the relational model, the conceptual model defines the *database schema* and is persisted into the *data dictionary* with the use of *Data Definition Language (DDL)* and *Data Manipulation Language (DML)*. ‘A database schema describes the database administrator’s [designer’s] knowledge of possible applications, the facts that can enter the database, or those of interest to end-users’ [154].

In The Linked Data Web, the conceptual model is defined using *vocabularies*, definitions for terminology-level classes and properties using RDF(S). RDF(S) databases are *schema-less*, where the vocabularies define the *descriptive* model of the domain, instead of a *prescriptive* model (in contrast to relational databases schemas). RDF(S) supports a dynamic evolution of the schema. The concept of vocabulary is based on the idea that some recurrent concepts and relationships in a specific domain of discourse can be formalized under a conceptual model which describes part of a domain. This agreement allows a level of conceptual model interoperability between different databases and between database and data consumer.

4.2.6 Semantic Web, Linked Data Web & Schema-agnostic Queries

The vision behind the Semantic Web is that the combination of principled knowledge representation approaches, added to large-scale data availability and logical inference mechanisms would support the level of semantic flexibility necessary to address semantic tasks such as semantic matching[68]. However, problems such as encoding data, logical inconsistencies at scale and scalability problems at large-scale dataset drove the simplification of the Semantic Web vision into the Linked Data Web[73], more aligned with a database perspective of semantics.

With the refocus from the Semantic Web to a Linked Data Web vision, the demand for semantic flexibility at query time increases with the growing availability of data, but the discussion on the aspects of data semantics decreased. From the Linked Data perspective, the semantic discussion was oversimplified to mechanisms such as the creation of vocabularies which play the role of shared terminologies or with the use or taxonomical inference mechanisms such as RDF(S).

Despite the major contribution of the Linked Data vision for providing an entity-centric data integration framework, from the perspective of data consumption, existing mechanisms in which Linked Data relies upon such as vocabularies and URIs have major associated costs in an open communication scenario and at their base rely on and are mediated by the interpretation of natural language descriptors.

4.3 Semantic Model for Databases

4.3.1 Motivation

This section revisits different perspectives on semantics aiming at providing a semantic model to support schema-agnostic queries. One demand of this discussion is to draw the line between the level of the semantic investigation that must be undertaken to support schema-agnostic queries and of semantic tasks which are dependent of a broader knowledge representation and artificial intelligence discussion.

At the limit, schema-agnostic queries are dependent on the sophistication of the semantic model behind them, which is strongly associated with knowledge representation and reasoning frameworks. While the investigation of more sophisticated semantic models, can improve the level of schema agnosticism, this work focuses on the investigation of semantic models to support the semantic matching between schema-agnostic queries and concepts in the database which are explicitly conceptualized.

4.3.2 Semantics: the Epistemological, Formal & Praxis perspectives

Semantic theories have its roots in the *epistemological thought*. From the evolution of epistemology, different theories on the nature of knowledge emerged, including the logical base developed by Aristotle, Leibniz, and Boole which culminated on the analytical philosophy in Frege and Russell. The approximation from analytical philosophy to mathematics, brought the increase on the emphasis on a increasing formalisation of

the logical thought. This tendency towards a formal perspective of semantics was reinforced by the growing number of connections between logics, computer science, artificial intelligence and computational linguistics.

However, the formal perspective on semantics concentrates on the analysis and description of specific semantic phenomena under simplifying and isolating conditions, creating a gap between the theoretical accounts and its applicability into complex and real world conditions [155]. As Baroni et al. [155] summarizes:

“Most semantic models have dealt with particular types of constructions, and have been carried out under very simplifying assumptions, in true lab conditions. If these idealizations are removed it is not clear at all that modern semantics can give a full account of all but the simplest models/statements.” [155]

This observation points into a third major perspective on semantics, the *praxis* perspective, which focuses on the proposal of approaches which can address the semantic phenomena at the level of complexity of its real world instances. These approaches are typically multi-disciplinary and are focused on a specific category of semantic tasks. The praxis perspective targets the construction of semantic models that can support the materialization as systems and resources. An exemplar system following the semantic praxis perspective is IBM Watson [130].

In order to cope with the complexity, semantic models from a praxis perspective focus on the following characteristics:

- *Focus on a specific category of tasks.*
- *Targeting real world data conditions.*
- *Effectiveness as best-effort/approximation, instead of targeting sound and complete models.*
- *Quality as the empirical measurement of performance for addressing a specific task.*

This work concentrates on providing a semantic model for schema-agnostic queries under the praxis perspective.

4.3.3 Requirements for a Semantic Model for Schema-agnostic Queries

Addressing schema-agnostic queries can be categorised as one instance of the ‘*[semantic] brittleness bottleneck*’ [46] as referred by Lenat, i.e. the symbolic rigidity of software

systems and databases which are dependent on a perfect symbolic and syntactic matching under the systems vocabulary. Addressing the brittleness bottleneck is intrinsically dependent on the ability to capture large-scale data and its semantic relations. According to Lenat [46], a solution for the brittleness bottleneck depends on addressing the following three tasks:

- Developing a declarative semantics language for knowledge representation.
- Developing a procedure for manipulating knowledge.
- Construction of the knowledge base (encoding the knowledge in the knowledge representation framework).

While these dimensions summarize the components necessary for semantic models to address most semantic problems, we claim that these components should be simplified in order to provide effective models for specific tasks such as the support for schema-agnostic queries.

The following requirements summarize the set of requirements for a semantic model for schema-agnostic queries. The requirements converges the tasks introduced by Lenat, a subset of the list of requirements for schema-agnostic queries, targeting a semantic model under the *praxis perspective*:

1. **Comprehensive Semantic Approximation Mechanism:** A semantic matching mechanism should be able to cope with the different *semantic mapping types* defined in Section 2.8.2. For the semantic matching task, semantic approximation is the final functionality and operation that the semantic model should support.
2. **Comprehensive Semantic & Commonsense Knowledge Base:** Open domain semantic approximation is dependent on large semantic and commonsense knowledge bases. The semantic model should be able to express large commonsense KBs.
3. **Low Knowledge Acquisition Effort:** There is a trade-off between the expressivity of a semantic representation formalism and the effort associated to acquire comprehensive commonsense knowledge bases. The semantic model should minimize the effort involved in encoding the commonsense knowledge into the KB.
4. **Justification:** The semantic matching mechanism should be able to provide a justification for the semantic matching. This supports the user for verifying the suitability of the answer.

5. **Generality/Transportability:** The semantic matching approach should be transportable to different domains and data models (inherited from *low setup & maintainability effort* requirement for schema-agnostic queries).
6. **Low Semantic Matching Execution Time:** The semantic matching should be able to address most of the matching operations in an interactive (< 10s) matching execution time (inherited from the *interactive query execution time* requirement for schema-agnostic queries).
7. **Scalability:** The semantic model should scale to large commonsense knowledge bases and for large databases (inherited from *scalability* requirement for schema-agnostic queries).

4.4 Semantic Models

There are different perspectives on semantics which evolved from *logics*, *linguistics* or *cognitive psychology*. These different views focus on describing different aspects of the semantic phenomena. In this section, these different perspectives of semantics are briefly analysed, in relation to their core characteristics and their connection with the semantic view behind databases, and how each perspective fit into supporting the requirements for a semantic model to allow the semantic matching for schema-agnostic queries.

4.4.1 The Formal (Logic) Perspective on Semantics

The formal/logical perspective of meaning evolved from the analytical philosophy program of providing a more rigorous representation which can support a more precise reasoning process. Logics interprets meaning as a calculus where knowledge is '*rigorous and explicit; modelled on methods in logic*' [156]. Most of the formal approaches are truth-conditional and model-theoretic, where the meaning of a sentence is taken to be a proposition which is true or false in relation to some model of the world [157]. The meaning of a proposition/expression are the instances (constants) in the model and predicates are functions from instances to truth-values.

Inference is part of the formal perspective of semantics, where the knowledge base is extended automatically by an algorithmic process. Querying and reasoning over a logical KB are highly dependent on logically consistent KBs.

The construction of large commonsense knowledge bases under a formal logic framework present major challenges: (i) lack of a built-in semantic approximation mechanism; (ii)

need for maintaining logic consistency in the KB, a requirement which is difficult to guarantee in large-scale commonsense KBs; (iii) performance and scalability problems for large KBs; (iv) data acquisition problems as new knowledge needs to be expressed under a formal representation and should not violate the consistency of the KB.

Relational, Linked Data and Semantic Web databases are grounded on a formal view on semantics based on first-order logics. The logic perspective of semantics is given priority under the computer science perspective as it provides the framework to analyse the soundness and completeness properties of different approaches.

4.4.2 The Cognitive Perspective on Semantics

The cognitive perspective on semantics emerged from investigations in human cognition and emphasizes the way humans form and use concepts and categories [149]. ‘*Categorization occurs in all sensory modalities ... providing the gateway between perception and cognition*’ [158] *apud* [149]. Categories are names for sets (predicates) that help in the organization of entities in the world and have a hierarchical structure. A category can have an associated concept, which is a description of the discriminating features of that object. A new object in the world is categorized according to these features or it will define a new category.

The concept which defines the meaning of a word used to describe the category is a lean representation of individual conceptualizations (that could transcend the level of description), which represents the social understanding for it. For example an aeronautical engineer has a very complex conceptualization for the word ‘*airplane*’, but still he can communicate with the layman interpretation of that word. Additionally, not all categories have associated words to describe it, relying on compositions of other categories.

The focus of cognitivist approaches to semantics is to provide models which explain how human cognition works, using some of the characteristics of these models as a basis for the creation of computational semantic models. In the next sections two semantic models which emerged from cognitive models are described: *prototypes* and *frames*.

4.4.2.1 Prototypes

The notion of category as a way to describe sets of objects has similarities with the predicates in the formal perspective of semantics. However, the latter perspective is shaped by the ‘*necessary and sufficient conditions*’, which states that a predicate is defined by a set of necessary conditions which are sufficient when considered jointly.

The necessary and sufficient conditions (NSC) model can be categorized by the following properties [149]:

1. Categorization depends on a fixed set of conditions or features.
2. Each condition is absolutely necessary.
3. The conditions are binary (true or false).
4. Category membership is binary (true or false).
5. Categories have clear boundaries.
6. All members of a category are of equal status.

The logical perspective on categories have been questioned in the context of prototype theory. Prototype theory is based on the notion that there are ‘better members’ of a category, changing the *binary membership* perspective to a *similarity degree* perspective. These ‘better examples’ of members of a category are called *prototypes*, and the membership of other objects with regard to a category can be determined by its similarity to a prototype. Different experiments in cognitive psychology were carried out to support this hypothesis, confirming that some categories have a ‘graded structure’ such as the category of birds and colors [159].

The prototype model of categorization contains the following characteristics (adapted from [149]):

1. Graded structure and membership (members of a category are not of equal status).
2. Prototypes as best examples.
3. No set of necessary conditions.
4. Family resemblance.
5. Fuzzy boundaries.

Prototypes introduce the notion of *prototype similarity* as a basic construct in the semantic model, providing a principled approach for conceptual approximation. Since the prototypes theory have concentrated more on the mechanisms to create concepts, the traditional notion of prototypes refer in many cases to extra-linguistic features as the category of colors [159] shows. Despite the fact that prototype theory is more focused on concept formation, some of its notions can be applied in a semantic model in the

context of schema-agnostic queries: for example semantic approximation as a built-in construct and the flexibilization of the *necessary and sufficient conditions* (NSC).

As a limitation, prototype theory does not have an explicit representation of the internal structure of the concepts, concentrating more on taxonomical structures

4.4.2.2 Frames

The theory of *cognitive frames* [158] claims that concepts in the human cognition are represented as frames, i.e. a structured description of a concept, which provides a basic set of attributes and values associated to a concept. In contrast to prototype theory, *frame semantics* provides an explicit representation of the internal structure of the concepts. Minsky [160] describes a frame as a cover term for ‘*a data-structure representing a stereotyped situation*’. According to Fillmore [161]² frame semantics assumes that the meanings of most words can best be understood on the basis of a semantic frame: a description of a type of event, relation, or entity and the participants in it.

Frames provide a higher level structured interpretation for a word based on higher level categories (Frame Elements) associated with words (lexical units).

According to Loebner [149] a frame can be defined as:

Definition 4.1 (Frame). A frame is a conceptual network of attribute-value assignments that fulfills the following uniqueness assumptions: *unique frame referent*, *unique attributes* and *unique values*.

Attributes have a functional role in the sense that they provide an unique value assignment for a value. The attribute can be seen under a relational perspective as it requires the referent to be associated with the value, generating an entity-attribute-value triple. Frame attributes can be classified according to four types:

- *Part attributes*: Provide a description of the mereology of an object.
- *Correlate attributes*: Specify objects of an independent existence to which the referent of the concept is uniquely related.
- *Property attributes*: Provide a description of the abstract properties for the referent (gender, nationality).
- *Event attributes*: Describes the events associated with the referent.

²<https://framenet.icsi.berkeley.edu/fndrupal/about>

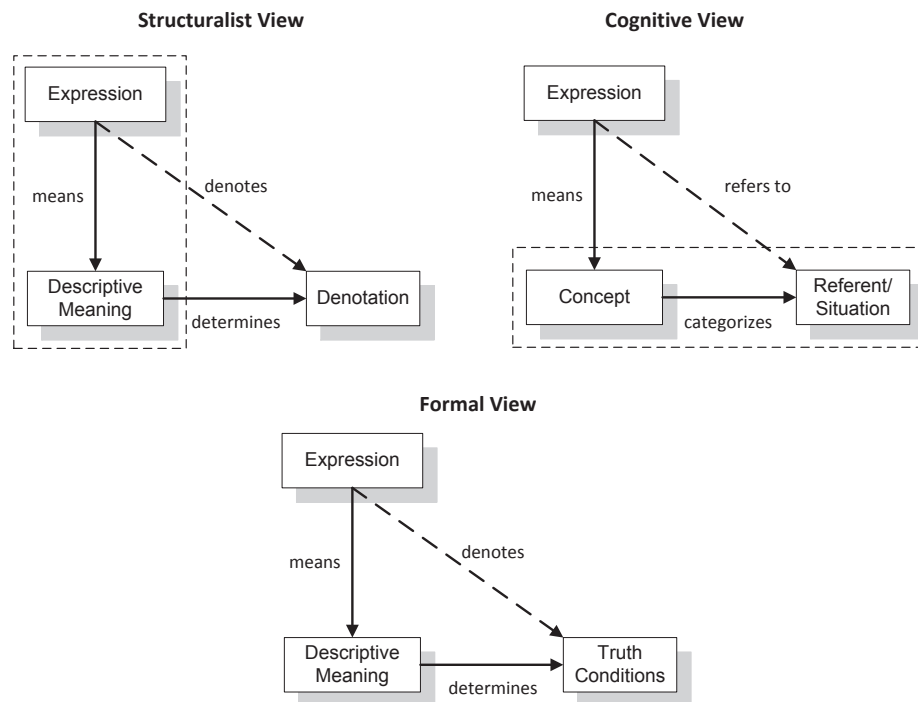


FIGURE 4.5: Semiotic triangle for the cognitive, structuralist and formal perspectives of semantics (Loebner, 2014 [149]).

According to [158] *apud* Loebner[149], frames and prototypes can be integrated through the specification of default values to prototypical properties.

Linked Data and Semantic Web databases (in the RDF(S) layers) share part of the perspective of meaning provided by frames, merging it with the logics perspective of meaning. Vocabularies and lightweight ontologies can be seen as mechanisms which provide a frame-based representation of concepts [162]. Similarly to the logical perspective, frame semantics depends on the explicit conceptualization of frames, increasing the data acquisition costs. However, frames concentrate on the representation of minimal models which describe the higher-level (taxonomical) categories associated with the structure.

4.4.3 The Structuralist Perspective on Semantics

The structuralist (Saussurrean) view of meaning evolved from a linguistic perspective: *‘the language is to be studied exclusively from within’* [149]. According to the structuralist view, the meaning of a sign is defined by its sets of relations and differences to the meaning of other signs (semantics as meaning relations). According to [149] *‘the structuralist notion of meaning is radically relational’*.

Dirk Geeraerts [163] summarizes the spirit behind the structuralist view of meaning:

“First, the study of meaning should not be atomistic but should be concerned with semantic structures. Second, it should be synchronic instead of diachronic, and third, the study of linguistic meaning should proceed in an autonomously linguistic way. Because the meaning of a linguistic sign is determined by its position in the linguistic structures of which it is a part, linguistic semantics should deal with those structures directly, regardless of the way in which they may be present in the individuals mind. Because the subject matter of semantics consists of autonomous linguistic phenomena, the methodology of linguistic semantics should be autonomous, too.” [163].

Saussure emphasizes that meaning arises from two kinds of differences between signifiers [164] *apud* [147]:

- *Paradigmatic*: Paradigmatic relations are functional contrasts, expressing differentiation. A paradigm is a set of associated signifiers or signifieds which are all members of some defining category, but in which each category is significantly different [147]. *‘Paradigmatic relations are those which belong to the same set by virtue of a function they share... A sign enters into paradigmatic relations with all the signs which can also occur in the same context but not at the same time’* [165].
- *Syntagmatic*: Syntagmatic relations are expressed by possibilities of combination. Syntagmatic relations refer to other signifiers co-present within a local context. *‘A syntagm is an orderly combination of interacting signifiers which forms a meaningful whole within a text’*. The study of syntagmatic relations reveals the conventions or *‘rules of combination’* underlying the production and interpretation of texts [147].

Whereas syntagmatic analysis studies the ‘surface structure’ of a text, paradigmatic analysis seeks to identify the various paradigms (or pre-existing sets of signifiers) which underlie the content of texts. Saussure noted that a characteristic of what he called ‘associative’ relations (i.e. paradigmatic relations) that such relations are held *‘in absentia’*, in the absence from a specific text of alternative signifiers from the same paradigm ([164] *apud* [147]). Structural analysis involves the analysis the existence of ‘underlying’ thematic paradigms (such as antonyms).

The structuralist perspective sees the semantics as a reflection from the relationships expressed in the text, where the syntagmatic relations define the context in which paradigmatic relations are defined. From an acquisitional point of view, the structuralist perspective can support approaches which are able to automatically extract semantic information from text.

More recently, the structuralist view have been empirically supported by strong evidence that it can support semantic models automatically built from syntagmatic/paradigmatic relations in the text, in particular in the context of *distributional semantic models*. The connection between structuralism and distributional semantics as a historical analysis is described by Sahlgren in [166]. According to Sahlgren [166]: “*The differential view on meaning that Harris assumes in his distributional methodology does not originate in his theories. Rather, it is a consequence of its theoretical ancestry. Although Harris’ primary source of inspiration was Bloomfield, the origin on the differential view on meaning goes back even further, to the cradle of structuralism and the Cours de linguistique generale. It is in this work [that] Ferdinand de Saussure lays the foundation for what will later develop into structuralism*”. For a further discussion the reader is referred to [166].

Section 4.5 provides a detailed analysis of the main elements of *distributional semantic models*.

While these models do not provide fine-grained semantic models, i.e. semantic models in which all elements are unambiguously defined, they can be used as approximative models for specific semantic tasks, in particular tasks which target semantic approximation based on linguistic knowledge.

4.4.4 Requirements Coverage

In the previous sections different perspectives on semantics were analyzed in relation to requirements for a semantic matching model for schema agnostic queries. Table 4.4.4 summarizes the requirements and which semantic models better covers each requirement dimension.

The structuralist/distributional perspective provides a semantic model which better covers the set of requirements. Since structuralist/distributional models are based on large-scale statistical linguistic evidence, they do not provide an easily interpretable justification mechanism for the semantic alignments which is convenient to be interpreted by human users.

From the time performance perspective, distributional semantics provides an approximative model in which its performance and scalability is dependent on the dimensionality of the distributional vector space. Techniques for reducing the dimensionality of the space are fundamental for the scalability of distributional models to large corpora.

4.5 Distributional Semantics

4.5.1 Introduction

Distributional semantics is built upon the assumption that the context surrounding a given word in a text provides important information about its meaning (*Distributional hypothesis*) [50]. A rephrasing of the *distributional hypothesis* states that words that occur in similar contexts tend to have similar meaning [50]. Distributional semantics focuses on the construction of a semantic representation of a word based on the statistical distribution of word co-occurrence in texts. The availability of high volume and comprehensive Web corpora [53] brought distributional semantic models as a promising approach to build and represent meaning. One of the major strengths of distributional models is from the acquisitional point of view, where a semantic model can be automatically built from a large text collection.

The distributional hypothesis can be interpreted under different levels of emphasis [167]:

Definition 4.2 (Distributional hypothesis). “*Words that occur in similar contexts tend to have similar meanings*” ([50]; [49]).

Definition 4.3 (Weak Distributional hypothesis). “*Word meaning is reflected in linguistic distributions. By inspecting a sufficiently large number of distributional contexts we may have a useful surrogate representation of meaning.*”

Definition 4.4 (Strong Distributional hypothesis). “*A cognitive hypothesis about the form and origin of semantic representations.*”

In the context of this work, the *weak distributional hypothesis* is assumed.

4.5.2 Distributional Semantic Models (DSMs)

Distributional Semantic Models (DSMs) represent co-occurrence patterns under a *vector space representation*. In this section, the core components of a distributional semantic model are described.

4.5.2.1 Distributional Vector Space

In DSMs, the meaning of a *word* is represented by a *weighted vector* where each dimension represents a *context* in which the word occurs in the corpora (Figure 4.6). A DSM defines a vector space for the set of words represented within the DSM.

A *vector space* is defined as:

Definition 4.5 (Vector Space). A real vector space $VS^{\mathbb{R}}$ is a set that is closed under finite vector addition ($V \times V \rightarrow V$) and scalar multiplication ($\mathbb{R} \times V \rightarrow V$), and should satisfy the following axioms (for vectors $\vec{u}, \vec{v}, \vec{w}$ and scalars a, b):

- For the vector addition:
 - *commutativity*: $\vec{u} + \vec{v} = \vec{v} + \vec{u}$
 - *associativity*: $\vec{u} + (\vec{v} + \vec{w}) = (\vec{u} + \vec{v}) + \vec{w}$
 - *identity*: $\vec{v} + 0 = \vec{v}$ for all $\vec{v} \in V$
 - *inverse*: for all $\vec{v} \in V$, exists $-\vec{v}$ such that $\vec{v} + (-\vec{v}) = 0$
- For the vector multiplication:
 - *associativity*: $(ab)\vec{v} = a(b\vec{v})$
 - *distributivity*: $a(\vec{u} + \vec{v}) = a\vec{u} + a\vec{v}$
 - *identity*: $1\vec{v} = \vec{v}$ for all $\vec{v} \in V$

Definition 4.6 (Linearly independent vector). n vectors $\vec{v}_0, \vec{v}_1, \vec{v}_2, \dots, \vec{v}_{n-1}$ are *linearly dependent* if exists n scalars c_0, c_2, \dots, c_{n-1} not all equal to zero such that: $c_0\vec{v}_0 + \dots + c_{n-1}\vec{v}_{n-1} = 0$. The vectors are *linearly independent* if this condition does not hold.

Definition 4.7 (Vector Space Basis). A basis for a vector space VS is defined as a subset of vectors

Definition 4.8 (Vector Space Dimension). The dimension of the vector space $VS^{\mathbb{R}}$ is the number of basis vectors in $VS^{\mathbb{R}}$.

DSMs are represented as a *distributional vector space*, where each dimension represents a *context* \mathcal{C} for the linguistic context in which the *target term* \mathcal{T} occurs in a reference corpora \mathcal{RC} .

Definition 4.9 (Target word). A *target word* t is the word in the reference corpora for which the distributional vector representation is generated.

Definition 4.10 (Context pattern). A *context pattern* is a linguistic pattern in the reference corpora \mathcal{RC} . A context pattern has an associated *context identifier* c .

The *distributional interpretation* of a target word is defined by a *weighted vector* of the contexts in which the word occurs, defining a *geometric interpretation* under a distributional vector space. The weights associated with the vectors are defined using an *associated weighting scheme* \mathcal{W} , which re-calibrates the relevance of more generic or discriminative contexts and normalizes the weighting of the vectors.

Definition 4.11 (Distributional vector for a target word). A *distributional vector* \vec{t} for a target word t is given by: $\vec{t} = w_0c_0 + w_1c_1 + \dots + w_{n-1}c_{n-1}$, if t co-occurs with c in the reference corpora and w_i are the weighting functions $\in \mathbb{R}$.

4.5.2.2 Context Patterns

Different context patterns can be defined for distributional semantic models, including *number of neighbouring word windows, sentences, paragraphs* and *documents*. Lexical categories and syntactic features (e.g. dependencies) are also used to define the context patterns.

Wider distributional contexts (e.g. paragraphs, documents), tend to capture syntagmatic relations (words with different meaning which frequently co-occur in the same context, such as *vehicle* and *wheel*, *war* and *weapon*) while narrow context windows will capture paradigmatic relations, i.e. words that occur in very similar syntagmatic contexts, typically *synonyms* and *antonyms*.

The example below shows an example of the target word ‘*child*’ / ‘*children*’ and the set of collocated words which define the context pattern in a small reference corpus. In this example, the context pattern is given by the set of *nouns* and *verbs* within a *context window* of five words.

Corpora:

... her **children** were *born* after the and his *family* including *wife* and **children**
 **sent** his **child** to *school* the **child** *played* and then she *went* to *school* ...

The process of selecting the best context, called *context engineering*, is dependent on its suitability of the task at hands and strongly affects the performance of the DSM.

4.5.2.3 Weighting Functions

Distributional models are built based on the frequency of co-occurrence patterns between target words and context patterns. Computed over a large reference corpora, higher co-occurrence frequencies may provide evidence that the context pattern is strongly associated with a target word. Some context patterns can be more discriminative and descriptive of the semantic content of the target word in relation to other context patterns. *Weighting functions* are used to recalibrate the weight of a context feature, weighting-up more salient contexts in comparison with other co-occurrence patterns. Common context patterns across different target words are weighted down. As Turney & Pantel [168] summarizes: ‘*the idea of weighting is to give more weight to surprising events and*

less weight to expected events'. The weighting function can also be interpreted as an automatic feature selection process.

Weighting functions are also important to normalize the size of context vectors, for example, document length normalization (Turney & Pantel [168]).

Different types of weighting functions can be applied [169]: *term frequency/inverse document frequency* (TF/IDF), *mutual information* (MI), *T-Test* among others are examples of different weighting schemes available from the literature. In case no weighting function is applied, $w_{ij} = f_{ij}$.

Definition 4.12 (Weighting function). A *weighting function* w_{ij} is a function of f , where f is the number of times that t_i co-occurs with the *context pattern* c_j in the *reference corpus* RC .

The example distributional vector for $\overrightarrow{\text{child}}$ over the example corpora is shown in Figure 4.6.

Example Vector:

- target word $\mathcal{T} = \text{'child'}$.
- context pattern $\mathcal{C} =$ nouns and verbs in the same sentence ($c_0 =$ school, $c_1 =$ born, $c_2 =$ family, $c_3 =$ wife, $c_4 =$ played, $c_5 =$ sent, $c_6 =$ went).
- weighting function $\mathcal{W} =$ frequency in \mathcal{C} .

The distributional vector for the target word is $\overrightarrow{\text{child}}$:

$$\overrightarrow{\text{child}} = \begin{pmatrix} \text{school} : 2 \\ \text{born} : 1 \\ \text{family} : 1 \\ \text{wife} : 1 \\ \text{played} : 1 \\ \text{sent} : 1 \\ \text{went} : 1 \end{pmatrix}$$

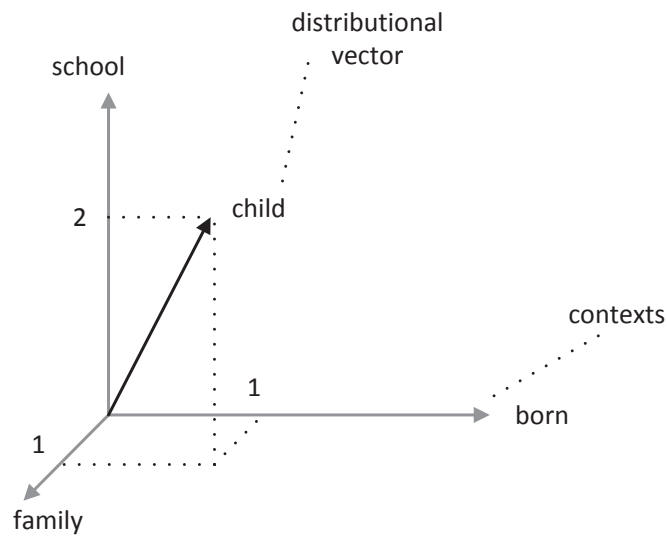


FIGURE 4.6: Depiction of the example of the distributional semantic representation of a word in a corpora.

		contexts									
		school	born	family	wife	cry	crawling	played	sent	went	...
target words	child	2	1	1	1	1	0	1	1	1	
	kid	3	1	1	1	1	1	1	1	1	
	baby	0	3	1	0	5	4	1	1	0	
	...										

FIGURE 4.7: Distributional matrix built from the context vectors of the target words.

4.5.2.4 Distributional Matrix

The set of distributional vectors for the target words can be organised into a *distributional frequency matrix* \mathcal{M} for the weights w_{ij} ($\mathcal{T} \times \mathcal{C}$), where the lines correspond to the target words and the columns to the context patterns (Figure 4.7).

The *distributional frequency matrix* consists of three steps: (i) sequentially scanning through the corpus, recording events of the type $\langle \text{target word}, \text{context vector} \rangle$, counting their occurrences, (ii) creating the table using a *sparse representation*, (iii) applying the weighting scheme [168].

4.5.2.5 Dimensionality Reduction

The number of distinct context patterns determines the *dimensionality* of the vector space. The dimensionality of the space has a strong computational impact on the performance of the distributional model. In order to address this problem, *dimensionality reduction* techniques are applied to reduce the dimensionality of the vector space. One example of dimensionality reduction operation is the Singular Value Decomposition (SVD) ([170], [171]), which computes a lower dimensionality matrix as an approximation of the higher dimensionality, minimizing the approximation errors. [170] and [171] interprets the dimensionality reduction process as a way to discover the *latent meaning* (each dimension corresponds to a latent meaning for different words).

4.5.2.6 Distance Measures: Semantic Similarity & Relatedness

The distributional semantics vector space contains a set of words represented by their weighted co-occurrence context patterns. According to the distributional hypothesis, words that contain similar contexts will tend to have similar meanings. As the vector space defines a *geometric representation* for the meaning of a word, words with similar contexts will tend to have vectors which are geometrically closer, in contrast with words with dissimilar contexts. This supports the definition of the correspondence between *geometric/vectorial distance* and *semantic similarity & relatedness*. In this case, the semantic similarity between two words t_1, t_2 is a function of their corresponding *vector distance* in $VS^{\mathbb{R}}$.

Two examples of *distance measures* are the *cosine similarity* and the *Euclidean distance*. The cosine similarity is defined as the angle between the two word vectors and it is calculated using the scalar product.

Definition 4.13 (Scalar product). Let $a = (a_0, a_1, a_2, \dots, a_n)$ and $b = (b_0, b_1, b_2, \dots, b_n)$ be vectors in \mathbb{R}^n . The *scalar product* is $a \cdot b$ is given by: $a \cdot b = \sum a_i b_i$

Definition 4.14 (Cosine similarity). Let $a = (a_0, a_1, a_2, \dots, a_n)$ and $b = (b_0, b_1, b_2, \dots, b_n)$ be vectors in \mathbb{R}^n . The cosine similarity of two vectors is given by the scalar product of two normalized vectors:

$$\cos(a, b) = \frac{a \cdot b}{\|a\| \|b\|}$$

The Euclidean distance measures the distance between the points defined by the word vectors and it is defined as:

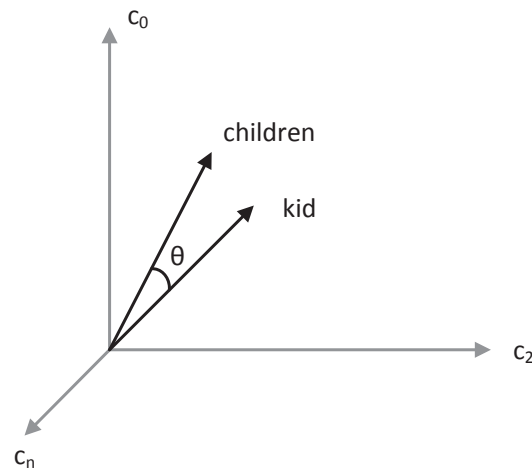


FIGURE 4.8: Depiction of the cosine similarity for the distributional vector space.

Definition 4.15 (Euclidean distance). Let $a = (a_0, a_1, a_2, \dots, a_n)$ and $b = (b_0, b_1, b_2, \dots, b_n)$ be vectors in \mathbb{R}^n . The Euclidean distance of the two vectors is given by the scalar product of two normalized vectors:

$$d(a, b) = \sqrt{\text{sum}(b_i - a_i)^2}$$

Other similarity measures can be defined [169]. Figure 4.8 depicts the *cosine similarity* for two example vectors.

4.5.2.7 Multiple Senses & Ambiguity

Most DSMs collect distributional evidence for all possible senses of a word into a single distributional vector. During the computation of semantic relatedness, the vector components (contexts) relative to the senses which match the other word sense are used in the computation of the semantic relatedness score. In this process, each word helps to select the best sense of the other word. This implicit word sense disambiguation process during the computation of the semantic relatedness measure penalizes the final score. However, in most cases the overlap of the matching sense contexts provides sufficient evidence that the two words are strongly related. Figure 4.9 depicts subspaces defined by two word senses.

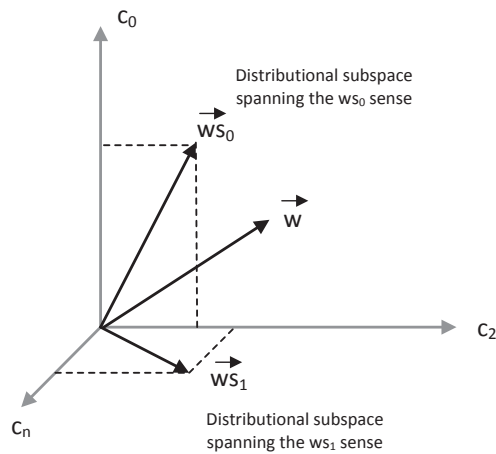


FIGURE 4.9: Depiction of word sense components for the distributional vector of a word.

4.5.2.8 Distributional Semantic Model

The following definition summarizes the core elements of a distributional semantic model.

Definition 4.16 (Distributional Semantic Model (DSM)). A Distributional Semantic Model (DSM) is a tuple $(\mathcal{T}, \mathcal{C}, \mathcal{R}, \mathcal{W}, \mathcal{M}, d, \mathcal{S})$, where:

- \mathcal{T} are the *target words*, i.e. the words for which the DSM provides a contextual representation.
- \mathcal{C} are the *context patterns* in which \mathcal{T} co-occur.
- \mathcal{R} is the *relation* between \mathcal{T} and the context patterns \mathcal{C} .
- \mathcal{W} is the *context weighting scheme*.
- \mathcal{M} is the *distributional matrix*, $\mathcal{T} \times \mathcal{C}$.
- d is the *dimensional reduction function*, $d : \mathcal{M} \rightarrow \mathcal{M}'$.
- \mathcal{S} *distance measure*, between the vectors in \mathcal{M}' .

This definition supports the understanding of which are the core elements of distributional semantic models, also supporting the classification of DSMs. Examples of different distributional semantic models are:

- Latent Semantic Analysis (LSA) [170].

- Random Indexing (RI) [172].
- Explicit Semantic Analysis (ESA) [53].

4.6 Effectiveness of Distributional Semantics: Semantic Similarity & Relatedness

4.6.1 Motivation

Due to the simplicity of its representation, distributional semantics enables the construction of comprehensive semantic models from large-scale unstructured text. The ability to extract semantic information from large scale corpora supports the construction of semantic models which addresses the *comprehensive semantic matching requirement* (Section 1.6).

However, the simplicity of the semantic representation implies that distributional semantic models are more coarse-grained in comparison to manually curated structured semantic models, restricting the scenarios in which distributional semantic models are effective.

The computation of *semantic similarity* and *semantic relatedness measures* is an important case in which the effectiveness of distributional semantics is empirically confirmed [53]. DS performs better than existing approaches based on structured and manually curated resources such as WordNet. Additionally, we argue that the effective computation of semantic similarity and relatedness measures should be first-class citizens in the process of mapping schema-agnostic queries to database elements.

The problem of measuring the *semantic similarity* and *relatedness* of two concepts can be stated as follows: given two concepts A and B , determine a numerical measure $f(A, B)$ which expresses the semantic similarity or relatedness between concepts A and B . The notion of *semantic similarity* is associated with taxonomic (is-a) relations, while semantic relatedness represents more general relations. ‘*Car*’ and ‘*train*’ are examples of similar concepts (both share a common taxonomic ancestor, ‘*vehicle*’) while ‘*car*’ and ‘*wheel*’ are related concepts (a wheel is part of a car) [51]. As a consequence, semantic similarity is considered a particular case of semantic relatedness.

Alternatively semantic similarity can also be defined as two concepts sharing a high number of salient features (attributes): synonymy (car/automobile), hyperonymy (car/vehicle), co-hyponymy (car/van/truck), while semantic relatedness [173] can be defined

as two words semantically associated without being necessarily similar: function (car/-drive), meronymy (car/tyre), location (car/road), attribute (car/fast) [149].

The problem of modelling and applying measures of semantic similarity and relatedness between two concepts has been investigated in different domains including cognitive psychology, artificial intelligence, information retrieval and natural language processing. Early approaches in cognitive psychology [51, 174] investigated semantic similarity and relatedness motivated by its centrality in the process of modelling the semantic memory in human cognition. Later, the cognitive semantic similarity and relatedness models were applied to the AI and computational linguistics domain.

However, for some years, the lack of structured semantic representations represented a barrier for its application in different semantic tasks. More recently, the availability of resources containing richer and more comprehensive structured semantic representations (thesauri, taxonomies, semantic networks) such as WordNet or ontologies, brought the investigation of semantic similarity and relatedness to a new phase. As a consequence, new measures based on linguistic resources such as WordNet and on ontologies were created.

The application of linguistic resources such as WordNet and ontologies lies at the core of existing approaches to address the vocabulary problem in its many instances, including schema-agnostic queries [118]. The use of these resources is not always mediated by the computation of associated similarity and relatedness measures, but also by simple synonym/hypernym/hyponym lookup approaches, ontology navigation algorithms and by taxonomical and logical inferences. The application of WordNet and ontology-based approaches for addressing schema-agnostic queries was previously analysed in Chapter 3.

The computation of semantic similarity and relatedness based on WordNet has the following limitations:

- *Limited number of concepts and relations*
 - *Lack of representation of non-taxonomic relations:* WordNet concentrates on the representation of taxonomical and synonymic relations, limiting the computation of semantic relatedness measures.
 - *Meaning evolution:* The use of words and their associated meaning is continuously evolving with new contexts of use. WordNet provides a snapshot of consensual meaning descriptions at a certain point in time. The evolution of the meaning of a word requires WordNet to be updated.

- *Meaning variation in more specific contexts:* Meaning can strongly vary in the context in which it is used. WordNet represents common and consensual senses that a word can be used: more specific or particular senses are not covered.
- *High construction effort/Low transportability:* WordNet was manually created by linguists. There is a high associated cost for updating WordNet to cope with meaning evolution, to cope with other languages or to cover domain-specific scenarios.

The limitations of WordNet brought to focus the use of approaches which could be automatically built from text. The availability of large amounts of unstructured text on the Web and, in particular, the availability of Wikipedia, a comprehensive and high-quality knowledge base [53], motivated the creation of similarity and relatedness measures based on these resources, focusing on addressing the limitations of WordNet-based approaches, trading structure for volume of commonsense knowledge [53]. Distributional semantic models were used to define new semantic similarity and relatedness measures. Distributional measures have shown clear improvements over previous WordNet-based approaches, getting closer to human-level assessments of semantic relatedness [53]. In addition, as observed by Gabrilovich & Markovich [53], the growth of reference corpora such as Wikipedia can represent a perspective of constant performance improvements of distributional approaches.

4.7 A Distributional Semantic Model for Databases

4.7.1 Distributional Grounding of Database Symbols

The analysis of the different perspectives of semantics shows that distributional semantics provides a semantic model which can address the requirements for the construction of a semantic matching mechanism for schema-agnostic queries. At the center of this strategy is the process of using a simplified vector-based semantic representation, which automates the semantic and commonsense knowledge acquisition effort and can support the process of semantic approximation between the query lexicon and the database lexicon.

Distributional semantics and the formal perspective of databases are complementary. The convergence between the existing database semantics and distributional semantics defines a hybrid model which inherits properties from different semantic perspectives (Figure 4.11). In the hybrid model, the crisp semantics of query terms and database elements is extended and grounded over a distributional semantics model, which is used in the semantic approximation process (Figure 4.12).

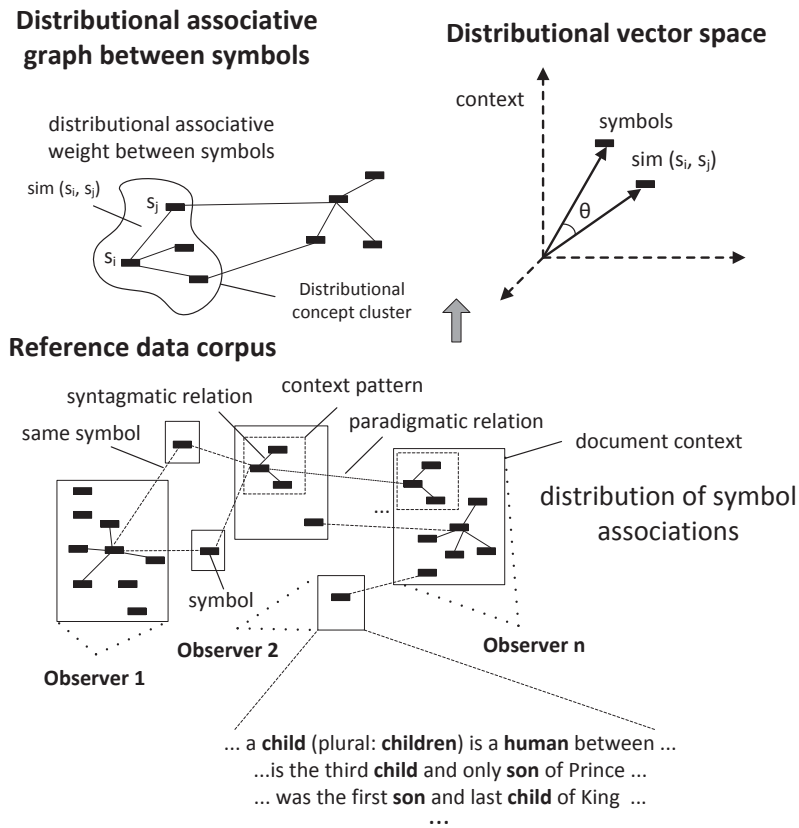


FIGURE 4.10: Depiction of distributional relations, contexts and different representation views for distributional semantics.

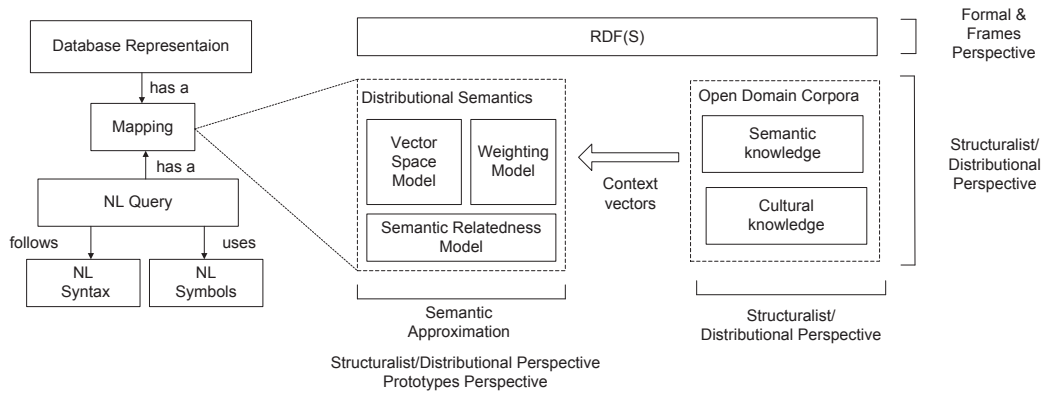


FIGURE 4.11: Semantic models of database elements.

The following definitions provide the core elements of the hybrid distributional-relational semantic model.

The alignment between a query term and a database term/element using a DSM is defined as a *d-alignment*.

Definition 4.17 (Distributional Semantic Model Alignment). Two terms t_1, t_2 have a

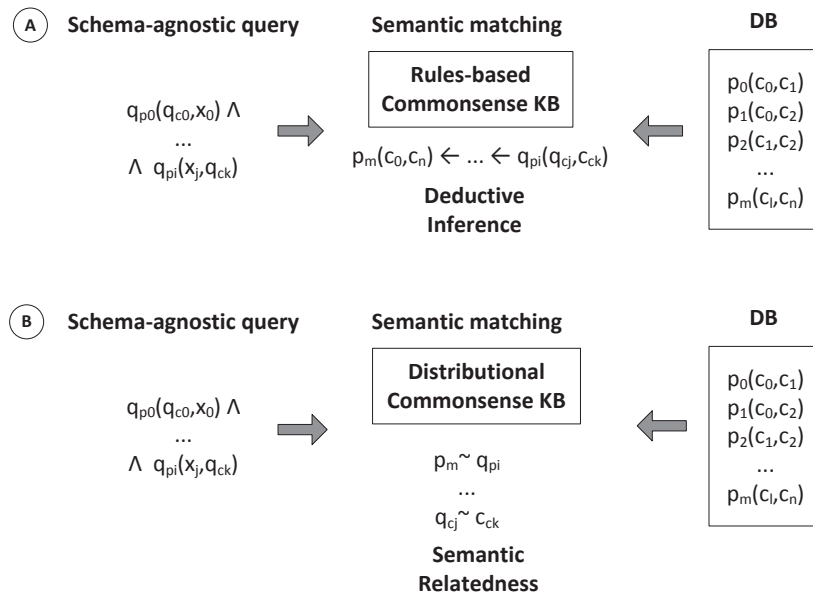


FIGURE 4.12: Logical (A) versus Distributional (B) alignment between query and database elements.

distributional semantic model alignment (d-alignment) if, for a distributional semantic model DM, they are semantically related according to a distributional model: $t_1 \sim^{DM} t_2$.

Definition 4.18 (Semantic Relatedness Score). Each *d-alignment* between two terms t_1, t_2 has an associated semantic relatedness score $s_{rel}(t_1, t_2)$.

Definition 4.19 (Semantic Relatedness Threshold). A *semantic relatedness threshold* η for a reference corpus \mathcal{RC} and a distributional model DM is a semantic relatedness value above which two terms are d-aligned. $t_1 \sim^{DM} t_2$ if $s_{rel}(t_1, t_2) \geq \eta$.

The semantic relatedness threshold defines a region in the vector space in which two terms can be considered semantically equivalent (Figure 4.13).

In order to introduce the discussion on the d-alignments between query and database, we start with a simplified example where the alignments are computed between a query term q and a set of database predicates p . The naive process of determining d-alignments consists of computing the distributional semantic relatedness scores between query terms and database terms, *ranking* the alignments according to their semantic relatedness values and filtering out alignments below the semantic relatedness threshold.

Figure 4.14 shows an example of a query and a database with a set of facts, while Figure 4.15 shows distributional semantic relatedness values between the query term ‘child’ and database predicates, ranked in a decreasing order. The final d-alignment is then determined (Figure 4.16) by applying the threshold η .

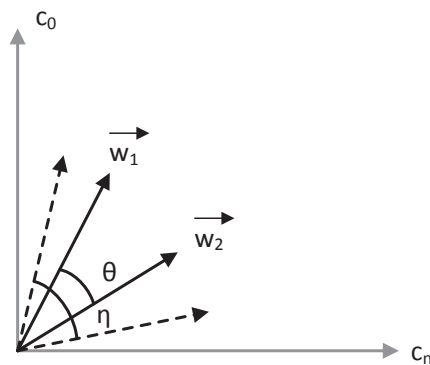


FIGURE 4.13: Depiction of the vector representation for the semantic relatedness threshold.

Query	DB
Who is the child of Bill Clinton?	almaMater (Bill Clinton, University College, Oxford) occupation (Bill Clinton, Politician) sonOf (Bill Clinton, Virginia Cassidy Blythe) religion (Bill Clinton, Baptists) fatherOf (Bill Clinton, Chelsea Clinton) ...

FIGURE 4.14: Example query and predicates.

Distributional Commonsense KB (Terminology-level)		
Higher semantic relatedness values	$s_{rel}(\text{childOf}, \text{fatherOf}) = "0.03259"$	$> \eta$
	...	$\eta = 0.02$
	$s_{rel}(\text{childOf}, \text{sonOf}) = "0.01091"$	
	$s_{rel}(\text{childOf}, \text{occupation}) = "0.00356"$	
	$s_{rel}(\text{childOf}, \text{religion}) = "0.00120"$	
Lower semantic relatedness values	$s_{rel}(\text{childOf}, \text{almaMater}) = "0.0"$	$< \eta$
	...	

FIGURE 4.15: List of database predicates for the example database ranked by their semantic relatedness score against a query term.

The computation of the distributional semantic relatedness is a semantic approximation process in which the *distributional knowledge serves as surrogate for the rules and axioms* in a commonsense knowledge base. The assumption is that the knowledge that would be expressed as rules and axioms is embedded in an unstructured way in the reference corpora, and that the *query-database provides the contextual and scoping mechanism in which the distributional knowledge can be applied as a semantic/commonsense approximation mechanism*.

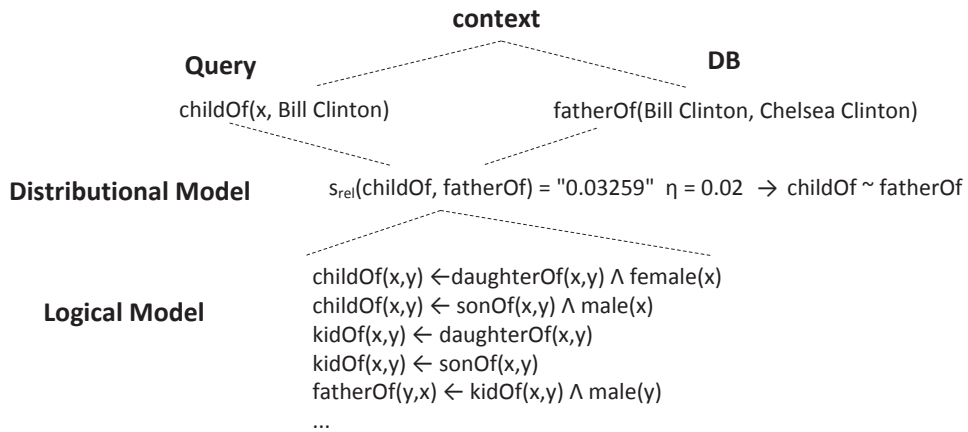


FIGURE 4.16: d-alignment between query term and database term.

An implication of the type $A(x) \leftarrow B(x)$ is equivalent to $B \subseteq A$. The d-alignment $A \sim^{\mathcal{DM}} B$ is not equivalent to $A(x) \leftarrow B(x)$ in absolute terms, i.e. for all possible inference contexts. However, we argue that given a certain *inference context*, the $A(x) \leftarrow B(x)$ implication can be *locally semantically equivalent* to a d-alignment. Figure 4.17 depicts the contrast between a rules-based inference approach and a distributional-based approach.

Definition 4.20 (Contextual Distributional Equivalence Hypothesis). Let t_q be a query term and t_{DB} be a database term. Let $\kappa(t_q)$ and $\kappa(t_{DB})$ be the contextual information associated with the query and database terms respectively (i.e. other words in the query and other entities in the database). If t_q is d-aligned with t_{DB} under the context $\kappa(t_q)$ and $\kappa(t_{DB})$ then t_q and t_{DB} can be assumed to be *semantically equivalent*.

The *contextual distributional equivalence hypothesis* is specialized in the scope of this work to assume that the query can provide sufficient contextual information for answering the query (i.e. the query is not intrinsically ambiguous or vague). This leads to the definition of the sufficient context condition:

Definition 4.21 (Sufficient Context Condition). The context $(\kappa(t_q), \kappa(t_{DB}))$ is said to be *sufficient* wrt to a distributional semantic model (DM) if there is a unique d-alignment $t_q \sim^{\mathcal{DM}} t_{DB}$

4.7.2 Semantic Best-effort

While structured query models target perfect accuracy models, the intrinsic semantic phenomena which emerges in a open communication scenario for schema-agnostic queries

Rules-based Commonsense KB (Terminology-level)	Distributional Commonsense KB (Terminology-level)
childOf(x,y) ← daughterOf(x,y) ∧ female(x)	s _{rel} (childOf, fatherOf) = "0.03259"
childOf(x,y) ← sonOf(x,y) ∧ male(x)	s _{rel} (childOf, sonOf) = "0.01091"
kidOf(x,y) ← daughterOf(x,y)	s _{rel} (childOf, kidOf) = "0.01046"
kidOf(x,y) ← sonOf(x,y)	s _{rel} (childOf, daughterOf) = "0.01059"
fatherOf(y,x) ← kidOf(x,y) ∧ male(y)	...
...	s _{rel} (childOf, occupation) = "0.00356"
	s _{rel} (childOf, religion) = "0.00120"
	s _{rel} (childOf, almaMater) = "0.0"
	...

FIGURE 4.17: Corresponding rules.

brings the demand to revisit the perfect accuracy expectations in this scenario. The approximative nature of distributional models brings an inherent level of uncertainty to the querying process.

In this scenario, the process of database querying should become closer to the information retrieval interaction approach, where users get a list of ranked results, but there is no expectation of perfect accuracy (precision = 1 and recall = 1). This is summarized in the *principle of the semantic best-effort*:

Definition 4.22 (Principle of Semantic Best-effort). In open communication schema-agnostic query approach scenarios, users should expect approximate results. Query mechanisms should maximize the precision and recall of the result set.

Given a sufficient context query set Q , the quality of the distributional model DM can be evaluated with regard to a database DB .

Definition 4.23 (Distributional Model Precision). The precision of a distributional model DM (d-precision) with regard to a database DB and a context-sufficient query set Q is given by:

$$precision(DM, DB, Q) = \frac{\text{number of } q_i \sim^{DM} t_j}{\text{total number of correct } q_i \sim^{DM} t_j}$$

Definition 4.24 (Distributional Model Recall). The recall of a distributional model DM (d-recall) with regard to a database DB and a context-sufficient query set Q is given by:

$$recall(DM, DB, Q) = \frac{\text{number of correct } q_i \sim^{DM} t_j}{\text{total number of correct } q_i \sim^{DM} t_j}$$

Given a pre-defined finite set of context-sufficient schema-agnostic queries Q , it is possible to define the conditions for a distributional model to satisfy a perfect accuracy condition. This implies in restricting the query set to a finite known query set (closing the query set). A DSM which can support this condition is called *d-separable* with regard to a query Q and a database DB .

Definition 4.25 (d-Separability). A DSM is *d-separable* for a finite set of schema-agnostic context-sufficient queries Q and for a database DB if *d-precision* = 1 and *d-recall* = 1 for all Q - DB d-alignments.

4.8 A Semantic Abstraction Layer for Databases

Distributional semantics provides a complementary perspective to the formal perspective of database semantics. While the formal perspective of meaning provides a crisp semantics for the closed communication (single context) scenario, distributional semantics can provide a flexible semantic layer for the open communication (multi-context) scenario. This layer supports schema-agnostic queries for databases for data environments under the SCoDD conditions. Similarly to the relational model which targeted creating a layer for abstracting users from the data management internals [66], distributional semantics supports the construction of a *conceptual/schema abstraction layer*, where users can be abstracted from the specific representation of the data. This complementary semantic abstraction layer grounds the semantics of the data in a vector representation over a reference corpora, which defines a generic architectural element for database management systems (DBMS) under the SCoDD conditions. As a *high-level architectural scheme*, the layer is represented in Figure 6.4.

Addressing the problem of schema-agnostic queries is dependent on a semantic model which supports the creation of large commonsense knowledge bases. Inference over commonsense knowledge bases can support the level of semantic approximation necessary to address the vocabulary problem for schema-agnostic queries. In the context of this work commonsense knowledge bases refer to the data and the semantic representation necessary to support the semantic approximation of query terms to database elements.

By simplifying the semantic representation using the vector representation, commonsense information can be automatically extracted from large-scale corpora, increasing the *completeness* of the commonsense KB in the spectrum of knowledge which is targeted by schema-agnostic queries, i.e. intensional linguistic knowledge (Figure 4.19). This supports a shift from the high-cost manual curation approaches which are used in the construction of semantic/commonsense knowledge bases and linguistic resources.

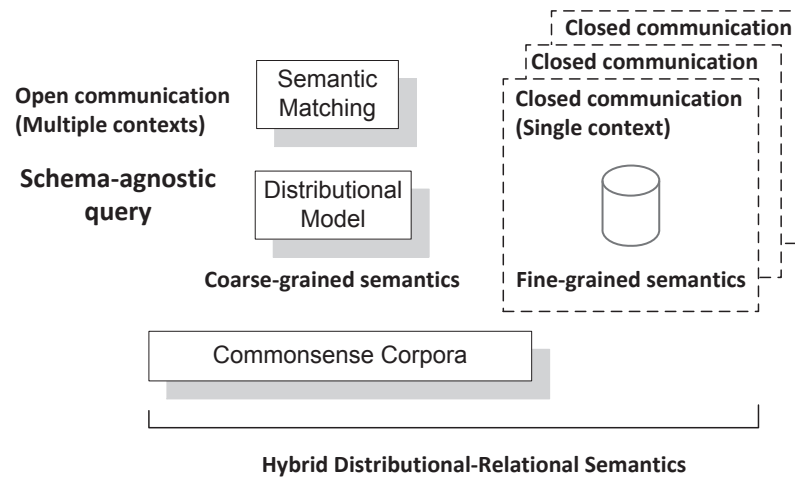


FIGURE 4.18: Distributional semantics layer complementing the database semantics.

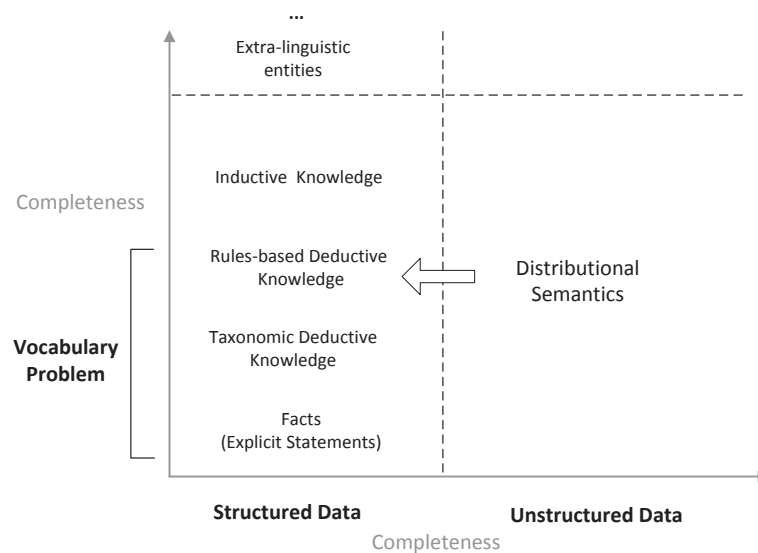


FIGURE 4.19: Semantic completeness of databases.

The proposed model can be seen as a step further from the need to formalize text domains into ontological structures for the problem of resolving schema-agnostic queries. Brewster, 2008 [175] proposes a revision of automatic ontology learning tasks towards making the task weaker (by acknowledging that perfect knowledge cannot be achieved in many cases) and by proposing a probabilistic framework in that each resource used in the process (the corpus, the extraction patterns and the extracted facts) can have an associated confidence level. This work takes this position one step further, by proposing that, for the task of resolving of schema-agnostic queries, the vector representation provided by distributional semantic models, provides sufficient semantic granularity to

address a large spectrum of semantic matching tasks, avoiding the need to extract a more formalized and structured model. Similarly, the semantic relatedness scores derived from the co-occurrence can serve as indicators confidence levels, which will demand in the future, its substitution by a more principled probabilistic measures as proposed in the Abraxas framework [175].

The conceptualisation, formalisation and empirical corroboration of the suitability of this semantic perspective for addressing the problem of semantic matching for schema-agnostic queries and its complementary nature to the formal database model is at the core of this thesis and its formalization ad evaluation is the focus of the following chapters.

4.9 Chapter Summary

At the center of the proposal of a schema-agnostic query approach is the definition of a semantic model which can cope with different *semantic mapping types*. This chapter provides a high-level analysis of the semiotic principles behind human-database communication and the associated semantic perspective on databases. Different perspectives on semantics (logical, cognitivist and structuralist) are analysed. Based on the analysis, a hybrid *distributional-relational semantic model* is outlined targeting to address the new semiotic assumptions which emerge in the *open communication scenario*. At the proposed model, distributional semantics is used to address the problem of semantic and commonsense data acquisition scale that is necessary for the construction of a semantic model to support schema-agnostic queries. The associated publications to this chapter are [176, 177].

Chapter 5

The Semantic Matching Problem: An Information-Theoretical Approach

“... to conduct my thoughts in such order that, by commencing with objects the simplest and easiest to know, I might ascend by little and little, and, as it were, step by step, to the knowledge of the more complex ...”

René Descartes, Discourse on the
Method

5.1 Introduction

The process of addressing schema-agnostic queries can be analyzed from an *information-theoretical* perspective, where the difficulty in addressing a query is proportional to the dimensionality of the *configuration space* associated with possible query-database alignments.

In this chapter, a preliminary information theoretical model for schema-agnostic queries is used to define measures of *semantic complexity* for matching *schema-agnostic queries*. The measures of semantic complexity can be used to quantify the role of central semantic phenomena such as *lexical and structural ambiguity*, *synonymy*, *vagueness* and the overall

matching complexity in the *semantic interpretation* of schema-agnostic queries. The quantitative model is then used in the design of the schema-agnostic query approach.

In order to achieve this goal, this analysis starts with an *abstract semantic matching model* (Section 5.2). The concept of entropy and its connection to semantic complexity is introduced in Section 5.3. From the two phases defined by the semantic matching model, a set of entropy measures are defined to quantify the different dimensions of uncertainty associated with each phase (Section 5.4). Section 5.5 defines the strategies to minimize the entropy in the semantic matching process.

5.2 Semantic Matching Model

In the *query-database semantic matching* two main categories of mapping processes can be distinguished:

- **Syntactic mapping:** The ability to align query terms to database elements according to valid query and database syntactic/structural constraints. Consists in the possible interpretations for the syntactic structure of the query under the database syntax. The entropy H_{syntax} expresses the *syntactic uncertainty/ambiguity* in the determination of the syntactic mapping.
- **Vocabulary mapping:** Corresponds to the semantic alignment between query terms and database elements. Consists in the matching/alignment between query terms and database entities. The entropy H_{vocab} is the *uncertainty/ambiguity* associated with the mapping between query terms and database entities.

These two processes are intrinsically intertwined as different lexical expressions can induce *different predicate-argument structures* (syntactic constraints).

In this work, the query-database semantic matching can be defined with the help of four sets: (i) a *word set* W , which expresses the set of words used to describe the domain of discourse shared by the query tokens and the database lexicon, (ii) a *word sense set* WS , which describes the possible senses associated of the words, (iii) a *composition set* S , to describe the possible (syntactically valid) compositions of words and (iv) a *concept set* C , to describe the set of concepts associated with the possible interpretation for all the compositions. The unambiguous *semantic interpretation* of a query $I(q)$ or database tuple $I(d)$ is a concept k_i in the concept set K . Figure 5.1 depicts the relationship between the sets in the query/database interpretation process.

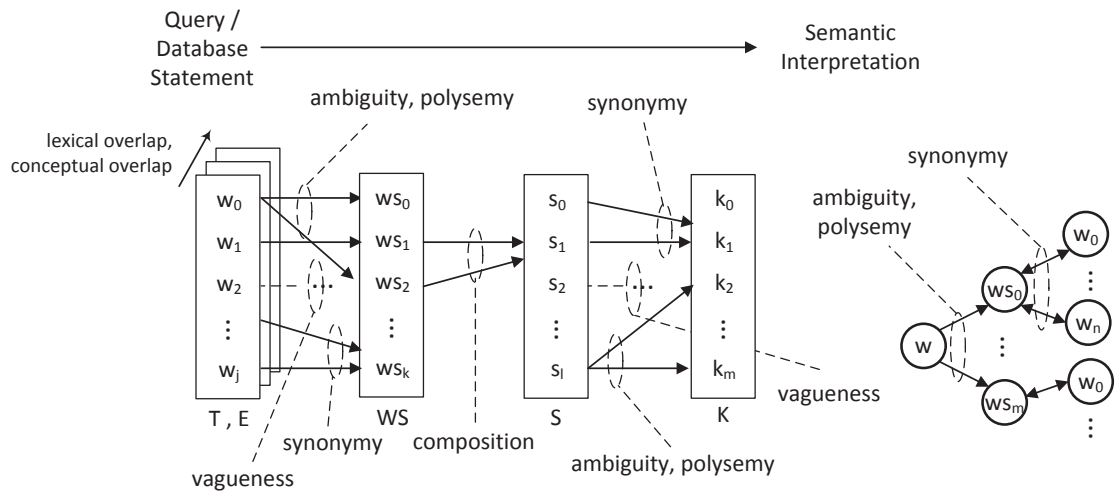


FIGURE 5.1: Abstraction reflecting the process of semantically mapping query to database elements.

In this model, lexical and syntactic ambiguity, vagueness and synonymy are defined as mapping patterns between the four sets (Figure 5.1).

A set M is defined for the *candidate mappings* between W and C under a specific query-database matching $m_\Lambda(Q, DB)$ under a *specific matching model* Λ . The *semantic entropy* associated with the query-database matching is proportional to the cardinality of M .

5.3 Semantic Complexity & Entropy

The concept of *entropy* in information theory is defined as a measure of uncertainty or surprise associated with a random variable. The random variable represents possibilities over the possible *states* or *configurations* that a specific symbolic system can be in, where the entropy is directly proportional to the number of states. In communication theory, the entropy is interpreted as the information content of a message between two communicating parties A (the receiver) and B (the transmitter).

Let X be a random variable with alphabet Ω and probability function $P(x), x \in \Omega$. Shannon [178] defines entropy based on probability terms as:

$$H(X) = \sum_{i=0}^n P(x_i) \log \frac{1}{P(x_i)}$$

where $P(x_i)$ is the probability of a symbol x_i occurring in a message. H provides a measure of the number of configuration states that the symbolic system (in this case the message) can be in. The higher the number of possible states and the more homogeneous the probability distribution, the higher the uncertainty on the state of the system.

In the context of schema-agnostic queries, this work interprets entropy under four main perspectives:

- (i) **structural/conceptual complexity:** Databases which express a large number of concepts have larger semantic entropy values. The number of possible query interpretations for a database is correlated to the number of distinct entities in the database and the number of possible compositions between them.
- (ii) **level of ambiguity:** Words/statements can convey different meanings. The degree of ambiguity (number of possible interpretations) varies for different words and propositions. Depending on the domain of discourse and on the selection of the words, queries and databases can have different levels of associated ambiguity.
- (iii) **vocabulary gap/synonymy/indeterminacy/vagueness:** Queries and databases may be expressed in different vocabularies or in different abstraction level and conceptual levels. Additionally, query and data may not be mapped with the contextual information available in the query or in the database (*indeterminacy/vagueness*).
- (iv) **novelty & informativeness:** Semantic entropy can be associated with the degree of *novelty/informativeness/surprise* associated with the communication process. The more informative the results returned to a query in relation to the background knowledge of the query issuer, the larger the entropy value. This dimension is not going to be the focus of this work.

Computing precise entropy measures for semantic matching is not always feasible due to the impossibility of precisely determining all the senses associated with a word. Although theoretical entropy measures can be defined, their application into a concrete query set or dataset would depend on an approximate model.

The next section introduces *semantic entropy measures* for each of the perspectives. In the definition of the entropies measures, an approximative perspective was adopted (which focuses on the computation of these measures instead of a purely formal model), where the definition of approximate measures take place wherever the complete model is not viable or practical.

A generic interpretation process for a schema-agnostic query Q can be defined as a set of steps which map a sequence of words $\langle w_0, w_1, \dots, w_n \rangle$ in the query Q into a set of possible database interpretations $I_{DB}(Q)$. Using the diagram in Figure 5.1, each configuration $I_{DB}(Q)$ can map to a concept in the concept set K . The interpretation of a query Q is a tuple $T = \langle C, P, R, L, Op \rangle$, where C and P are the set of constants and predicates in the database, $R \rightarrow P \times C \times \dots$ is the ordered set of syntactic n -ary associations between C and P , L is the set of logical operators \wedge, \vee and Op a set of functional operators. It is assumed that both query and database terminologies are defined under the same language and that database entities are described using natural language labels. In this section, to maximize generalizability the logic (constant, predicate) terminology is used to express database statements and queries.

There is no single generic process of interpreting the query against the database. A high-level query interpretation workflow is used to provide an association between query interpretation and semantic entropy measures. The query workflow consists of two steps (which maps to the two main categories of mapping processes): (i) *syntactic interpretation & predicate-argument structure determination* and (ii) *entity matching*.

Figure 5.2 depicts the steps in the query interpretation process and the associated entropies categories (described in the following sections), while Figure 5.3 depicts an example for a specific query example.

5.4 Measures of Semantic Entropy

5.4.1 Syntactic Entropy (H_{syntax})

The *syntactic entropy* of a query is defined by the possible syntactic configurations in which a query can be interpreted individually or taking into account the database syntax (predicate mapping). Figure 5.2 and Figure 5.3(2) depicts H_{syntax} within the query interpretation model. Let n be the number of words in the query Q .

One component of syntactic entropy is how the query can be segmented into terms which will map to database entities. We define the number of segments $N_{seg}(W)$ as the number of possible groupings between adjacent words for a query string W . The probability associated with a specific segmentation is:

$$P_{seg}(W \rightarrow Q) = \frac{count(W \rightarrow Q)}{N_{seg}(W)}$$

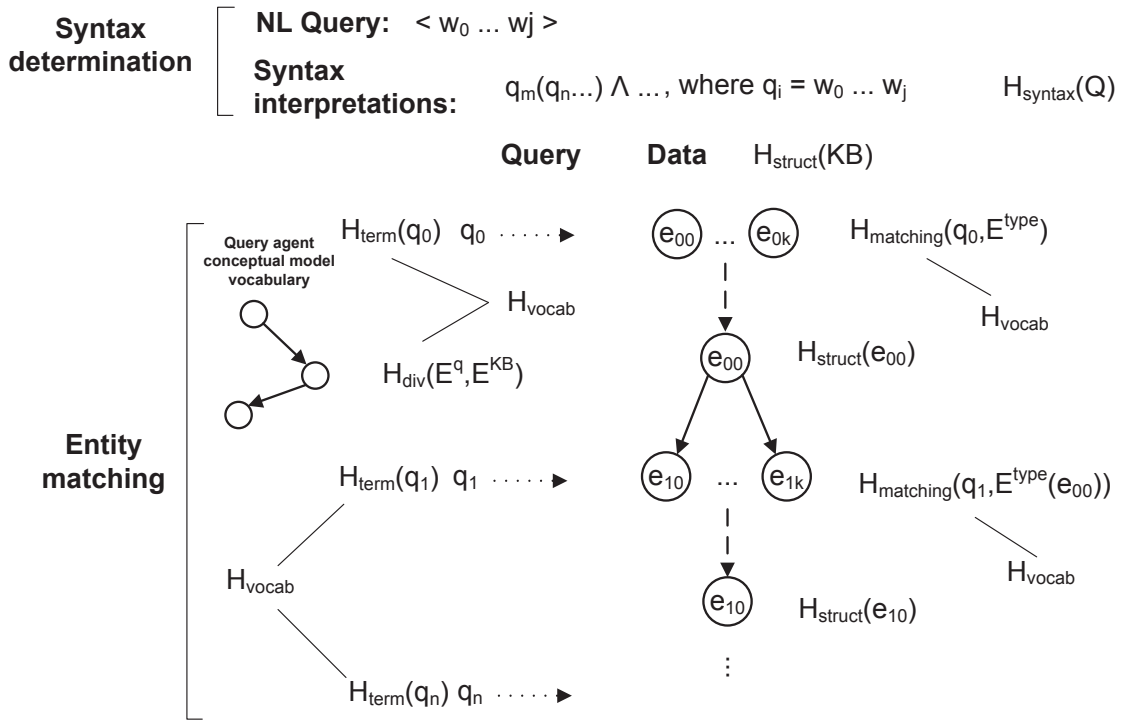


FIGURE 5.2: Generic steps for the query processing and associated entropy measures for each step.

where $\text{count}(W \rightarrow Q)$ is the number of observed instances that the word sequence W was segmented into the term sequence Q .

Another entropy component is associated with the syntactic parsing associated with the query terms. Let Syn be the *lexical categories* and *constituent categories* associated with the set of query words w_i and terms q_i . Let $N_{\text{cat}}(q_i)$ be the number of possible categories Syn associated with a query term q_i and $\text{count}(q_i \rightarrow Syn)$ the number of observed instances of the mapping $\text{count}(q_i \rightarrow Syn)$. The probability of a term q_i categorization is given by:

$$P_{\text{cat}}(Syn|q_i) = \frac{\text{count}(q_i \rightarrow Syn)}{N_{\text{cat}}(q_i)}$$

Given a segmented query Q^{seg} , the probability of a categorization is:

$$P_{\text{cat}}(Q^{\text{seg}}) = \prod_{i=0}^N P(Syn|q_i)$$

The overall entropy of a syntactic parsing is given by:

$$H_{syntax}(Q) = \sum_{Q^{seg} \in Seg(Q)} P_{cat}(Q^{seg}) \log P_{cat}(Q^{seg})$$

where the set of possible segments for a query Q is represented by $Seg(Q)$:

5.4.2 Structural Entropy (H_{struct})

The *structural entropy* defines the complexity of a database based on the possible statements that can be encoded under its schema. It provides a numerical description of amount of the information expressed in the database, independently of the query. Pollard & Biermann [179] proposed a structural entropy measure to quantify the entropy of a structured database. The entropy is computed by taking into account the number of database entities and their syntactic combination. Figure 5.2 and Figure 5.3(5,8,12) depicts H_{struct} .

Let's assume a database with a data model DM with two data model categories: constants $c \in C$ and predicates $\pi \in \Pi$. Let $t \in T$ be the set of tuples in the database containing constants and predicates. The probability of a constant ($P_{struct}(c)$) in the database is given by:

$$P_{struct}(c) = \frac{\mu(c)}{count(T)}$$

where $\mu(c)$ is the cardinality function.

$$\mu(c) = \begin{cases} 1 & , \text{ if } c \in T \\ 0 & , \text{ if } c \notin T \end{cases}$$

where $count(T)$ is the number of tuples in the database.

The probability of a predicate ($P_{struct}(\pi)$) in the database is given by:

$$P_{struct}(\pi) = \frac{\mu(\pi)}{count(T)}$$

where $\mu(\pi)$ is the cardinality function.

$$\mu(\pi) = \begin{cases} 1 & , \text{ if } \pi \in T \\ 0 & , \text{ if } \pi \notin T \end{cases}$$

where $count(T)$ is the number of tuples in the database.

The entropy of an entity $e \in E$ (either constant c or predicate π) is defined by:

$$H_{struct}(DB) = - \sum_{e \in E} P_{struct}(e) \log P_{struct}(e)$$

5.4.3 Terminological Entropy (H_{term})

The *terminological entropy* focuses on quantifying an estimate on the amount of *ambiguity*, *synonymy* and *vagueness* for the query or database terms independently of the matching between each other. The terminological entropy is proportional to the number of possible senses that a word may express. It provides a prospective measure of the semantic matching complexity by the query or database terminology itself.

Two types of word sense distributions are taken into account:

5.4.3.1 Uniform Distribution

The probability of w expressing a sense ws $P_{sense}(w \rightarrow ws)$ is given by:

$$P_{sense}(w \rightarrow ws) = \frac{1}{N_{sense}(w)}$$

where $N_{sense}(w)$ is the number of senses for w and an uniform probability distribution is assumed. $1 - P_{sense}(w \rightarrow ws)$ provides the probability that the word w expresses a different sense in an arbitrary context.

The probability of a concept ws being expressed in a particular lexical form w , assuming an uniform probability distribution.

$$P_{syn}(ws \rightarrow w) = \frac{1}{N_{syn}(ws)}$$

The *terminological entropy* can be defined by the *number of synonyms* and *number of senses* dimensions and are expressed as:

$$H_{term}^{syn}(ws) = \sum_{\forall ws \in WS} P_{syn}(ws \rightarrow w) \log \frac{1}{P_{syn}(ws \rightarrow w)}$$

$$H_{term}^{sense}(w) = \sum_{\forall w \in W} P_{sense}(w \rightarrow ws) \log \frac{1}{P_{sense}(w \rightarrow ws)}$$

where WS and W are the number of senses and words in a specific system (query or database).

5.4.3.2 Reference Distribution

Since the distribution of word sense and synonyms is not uniform we can assume that it is possible to estimate an approximation in a target domain.

In this case, the probability of w expressing a sense ws , $P_{sense}(w, ws)$ is given by:

$$P_{sense}(w \rightarrow ws) = \frac{N_{corp}(w \rightarrow ws)}{N_{corp}(w)}$$

where $N_{corp}(w)$ is the number of instances of words in a reference corpus and $N_{corp}(w \rightarrow ws)$ is the number of occurrences of w such that $w \rightarrow ws$ in a reference corpus.

The probability of a concept ws being expressed in a particular lexical form w .

$$P_{syn}(ws \rightarrow w) = \frac{N_{corp}(ws \rightarrow w)}{N_{corp}(ws)}$$

where $N_{corp}(ws)$ is the number of occurrences of the concept ws in the reference corpus and $N_{corp}(ws \rightarrow w)$ is the number of occurrences where ws is expressed as w .

The associated *terminological entropy* values for a non-uniform distribution are:

$$H_{term}^{syn}(ws \rightarrow w) = - \sum_{\forall ws \in WS} P_{syn}(ws \rightarrow w) \log P_{syn}(ws \rightarrow w)$$

$$H_{term}^{sense}(w \rightarrow ws) = - \sum_{\forall w \in W} P_{sense}(w \rightarrow ws) \log P_{sense}(w \rightarrow ws)$$

where WS and W are the number of senses and words in a specific system (query or database).

Different strategies can be used to build approximations for the terminological entropy. One example on approximate terminological entropy measure is the *translational entropy* [180] which uses the coherence in the translation of a word (translational distribution) as an entropy measure. Given a set of ordered word pairs (s, t) , respectively coming from a source language and a target language, an iterative process is used to determine the frequency $F(s, t)$ in which a word s is translated to a word t where $F(s)$ is the

absolute frequency of the source word in the text. The probability that s translates to t is defined as $P(t|s) = F(s,t)/F(s)$. The notion of probability is defined by the translational distribution, the term $H(T|s)$ is generated, calculating the entropy of a given word s against the target words set T :

$$H_{trans}(T|s) = \sum_{t \in T} P(t|s) \log \frac{1}{P(t|s)}$$

Figure 5.2 and Figure 5.3(6,10) depicts H_{term} .

5.4.3.3 Matching Entropy ($H_{matching}$)

Consists of measures which describe the uncertainty involved in the query-data matching/alignment between *query terms* and *database entities*. While terminological entropy measures provide an isolated estimate of the entropy, providing a prospective estimate of the matching complexity, the query-data entropy matching provides an estimate based on the set of potential alignments.

Two cases are considered depending on the distribution of word senses in a domain.

5.4.3.4 Uniform Distribution

Given a word w and a database DB with n_{DB} tuples, the probability of w matching a tuple in the database containing the same sense in the database $P_{sense}(w)$ is given by:

$$P_{sense}(w \rightarrow ws) = \frac{1}{N_{DB}(w)} \times \frac{1}{N_{sense}(w)}$$

where $N_{DB}(w)$ is the number of tuples in the database containing the word w and $N_{sense}(w)$ is the number of senses of the word w . In this example, a uniform distribution for the occurrence of the word senses is assumed.

In order to estimate the matching probability, the presence of possible synonymic terms should be taken into account. The total matching probability for a concept ws expressed in the query is given by:

$$P_{matching}(ws \rightarrow w) = \frac{1}{N_{syn}(ws)} \times P_{sense}(w \rightarrow ws)$$

where the $N_{syn}(ws)$ is the set of synonyms for a concept ws .

Defining Q^{ws} as the set of senses expressed in the query Q .

$$H_{matching}(Q^{ws}) = \sum_{\forall ws \in Q^{ws}} P_{matching}(ws \rightarrow w) \log \frac{1}{P_{matching}(ws \rightarrow w)}$$

5.4.3.5 Reference Distribution

Since the distribution of word sense and synonyms in the world is not uniform we can assume that we can know its approximation in the domain of discourse of the database and the query agent. Thus, the matching probability becomes:

$$P_{matching}(w \rightarrow ws) = \frac{1}{N_{DB}(w)} \times P^C(w \rightarrow ws)$$

$$P_{matching}(ws \rightarrow w) = \frac{\sum \forall w \in Syn(w) \times P^C(w \rightarrow ws)}{N_{Syn}(ws)}$$

where: $P^C(w \rightarrow ws) = \frac{N_C(w \rightarrow ws)}{N_C(w)}$

$$H_{matching}(Q) = \sum_{\forall ws \in Q} P_{matching}(ws \rightarrow w) \frac{1}{P_{matching}(ws \rightarrow w)}$$

Figure 5.3(4,7,11) depicts the $H_{matching}$.

5.4.3.6 Background Knowledge Entropy (H_{div})

Different parties interacting in a query-database scenario have different conceptualizations of the reality, vocabularies and distinct background knowledge. This difference in the background knowledge affects the ability of a querying agent to interpret the schema in which the data is expressed (or the ability of a database to interpret a query) and at the same time it may provide an indication of the amount of novelty in the database in relation to the query agent. *Kullback-Leibler (KL) divergence* can be adapted to provide a measure of similarity between the probability distributions of words between the conceptual models of a query agent(C_Q) and a database(C_{DB}).

The KL divergence $H_{div}(C_Q||C_{DB})$ is defined by the comparison of the probabilities of the distribution of words and word senses in the two conceptual models of $P_w(C_Q)$, $P_{ws}(C_Q)$ with regard to $P_w(C_{DB})$, $P_{ws}(C_{DB})$. As a condition for being computed, the

Measure	Semantic Measure Category	Type	Semantic Phenomena	Application
Pollard & Biermann [179]	Structural	Precise	Possibilities	Query-Data Alignment or Data
Translational Entropy (Melamed [182])	Terminological	Approximate	Ambiguity, Synonymy, Vagueness	Query or Data
Distributional Entropy	Terminological	Approximate	Ambiguity, Vagueness	Query or Data
Matching Entropy	Terminological	Approximate	Ambiguity, Vagueness	Query-Data Entity Alignments
Kullback-Leibler divergence (Term/Distributional)	Background Knowledge	Approximate	Novelty, Interpretability	Query or Data

TABLE 5.1: Classification of entropy measures according to associated features.

distributions should be defined in the same sample space and the distribution probabilities need to add up to one. Since the terminologies between C_Q and C_G are not likely to coincide, the smoothing approach suggested by Bigi [181] is used (which defines a very small weight value to represent the frequency of the words that are not present in the other conceptual model).

$$H_{div}(C_Q||C_G) = \sum_i^n P_w(C_Q) \log\left(\frac{P_w(C_Q)}{P_w(C_G)}\right)$$

In this work we define an extension to the KL measure to take into account distributional semantic concept vectors. The distributional Kullback-Leibler (KL) divergence, instead of using a sample space based on the words, it uses the distributional context vectors associated with the query and data terms. The context vector distribution for the conceptual model of the query and of the database is given by:

$$H_{div}^\kappa(C_Q||C_G) = \sum_i^n P_\kappa(C_Q) \log\left(\frac{P_\kappa(C_Q)}{P_\kappa(C_G)}\right)$$

A summary of the different entropy measures and their mapping to the associated semantic phenomena can be found in Table 5.1.

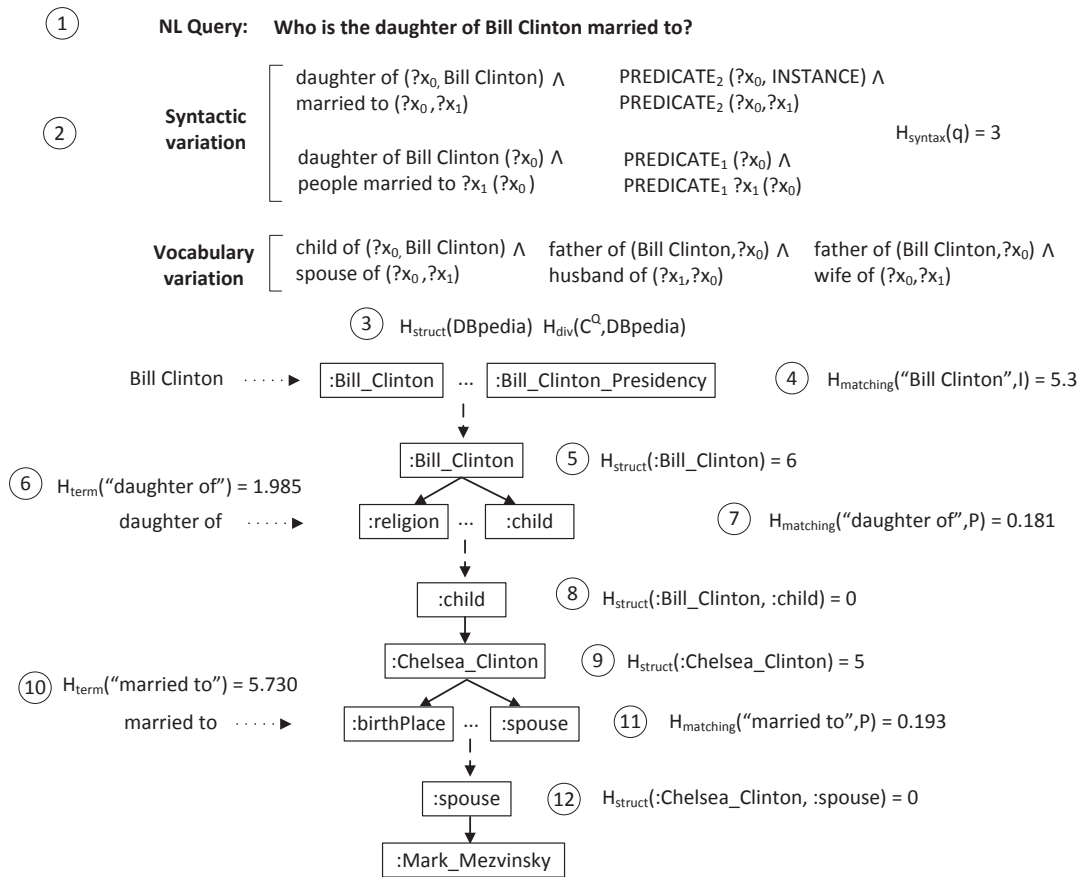


FIGURE 5.3: Instantiation of the query-entropy model for the example query.

5.5 Minimizing the Entropy for the Semantic Matching

The process of schema-agnostic query processing consists in the *minimization of the semantic entropy* associated with each step of the semantic matching process. Three types of *semantic matching strategies* can be distinguished depending on how the *syntactic matching* combines with the *entity matching* component.

1. *Pure Entity Matching*: where the explicit syntactic/structural information is ignored.
2. *Entity Matching followed by Syntactic Matching*: where the vocabulary alignment constraints are prioritized over the syntactic/structural constraints.
3. *Coupled Entity-Syntactic Matching*: where vocabulary and syntactic/structural matching steps are alternated.

5.5.1 Semantic Pivoting

In all the semantic matching strategies the first alignment is given by an entity matching step. The advantage of the coupled vocabulary-syntactic matching method in relation to the other two methods is that it supports the use of previous alignments together with syntactic constraints for the *reduction of the configuration space for the entity matching*, reducing the impact of the *structural entropy* H_{struct} , *terminological entropy* H_{term} and *vocabulary matching entropy* H_{vocab} components in the vocabulary matching process.

An entity matching which constrains the following alignments via the consideration of the syntactic/structural constraints is defined in the context of this work as a *semantic pivot*. The semantic pivot provides a *semantic context* for the next query-database alignments.

Definition 5.1 (Semantic Pivoting). The *semantic pivoting* operation consists of the minimization of the uncertainty associated with the first matching followed by the reduction of the semantic entropy for the following matchings.

Semantic pivots can be both constant (named entity)-type words or predication-type words.

The *first semantic pivot* plays a primary role in the query-dataset matching as it needs to cope with the full entropy of the query-dataset alignment. In order to maximize the alignment accuracy, an *heuristics* for selecting the alignments with the lowest entropies is fundamental. Named entities and their associated constant (instance)-type alignments have (in most scenarios) lower levels of vagueness, ambiguity and synonymy in comparison to predicate-type alignments, reducing H_{vocab} and H_{term} .

Other heuristics can be applied for selecting alignments with lowest entropy levels, and among predicate-type entities. The entropy minimization heuristics based on the selection of the semantic pivot are discussed in Section 8.5.3.4.

The semantic entropy reduction for an alignment using a semantic pivot produces a drastic reduction of the *configuration space*, if we take into account that the query syntactic constraints and the structural constraints in the database should be respected (the next entity alignment should be connected to the semantic pivot up by relationships entailed by a tuple). This is an assumption which is important in the context of schema-agnostic database queries: that the relationships between query terms can be expressed in the database using different conceptualisations, but the relationship is explicitly stated (in contrast to a possible inductive approach).

In most databases, the number of constants is much larger than the number of predicates. This implies that choosing a constant as a semantic pivot provides a larger reduction of the configuration space associated with the semantic matching. For a sequence of query-database alignments, the probability of two query terms $\langle q_i, q_{i+1} \rangle$ matching a database tuple $\langle e_j, e_{j+1} \rangle$ is given by:

$$P(q_{i+1} \rightarrow ws | q_i \rightarrow e_j) = \sum_{\forall w \in N_{syn}(ws)} P_{sense}(q_{i+1} \rightarrow ws)$$

$$P_{sense}(q_{i+1}) = \frac{1}{N_C(e_j)} \times \frac{1}{N_{sense}(q_{i+1})}$$

where $q_i \rightarrow e_j$ is the pivot alignment and the $N_C(q_i)$ number of tuples containing e_j candidates.

The last two equations show the strong dependency between $N_C(e_j)$ and $P(q_{i+1} \rightarrow ws | q_i \rightarrow e_j)$. Without the semantic pivot, the $N_C(e_j)$ term would be substituted by the number of tuples T in the database, where in most cases $T \gg N_C(e_j)$.

5.5.2 Syntactic Matching

The selection of different concepts to express semantically equivalent statements, as in Figure 5.3, strongly impacts the ability of relying on a rigid predicate-argument configuration in a schema-agnostic scenario. Different concepts expressed in the query define different syntactic constraints. Typically, logic-based QA approaches use the predicate-argument syntactic structure derived from the query lexicon as a rigid structure in which the database information should fit. We argue that in a schema-agnostic scenario, the entity matching precedes in priority the syntactic matching, i.e. the syntactic structure is dependent on the set of concepts/words used to express the data.

5.6 Chapter Summary

This chapter provided a *quantitative information-theoretic analysis* of the semantic complexity associated with matching schema-agnostic queries. The core goal of the chapter was to provide a *quantitative model* for schema-agnostic query-database matching. Different entropy measures corresponding to different dimensions of semantic entropy are defined, and approximative models based on literature work are proposed when exact models are not feasible to be calculated. The analysis of the entropy measures indicate

a substantial reduction of the matching entropy with the use of a *semantic pivot-based model*, in which elements with lower semantic matching entropies are resolved first, providing a context-based reduction mechanism of the entropy values for the remaining mappings. The associated publications to this chapter are [121, 183].

Chapter 6

$\tau - Space$: A Hybrid Distributional-Relational Semantic Model

*“Per ora, io vorrei codificare
l’incodificabile.”*

Leo Ferré

6.1 Introduction

In Chapter 4 the motivation and the main principles behind the development of a database semantic model based on distributional semantics were introduced. This chapter deepens this discussion, introducing the semantic model for supporting schema-agnostic queries, focusing on the complementary aspects between the *distributional semantics* and the *relational/logic-based* models perspectives. A hybrid distributional-relational model, named $\tau - Space$ is introduced as a data representation framework which unifies these two perspectives, where the relational/graph structure provides the *fine-grained and contextual semantic model*, which is complemented by the distributional model, which works as a *large-scale coarse-grained semantic/commonsense semantic model*. The $\tau - Space$ provides a principled and built-in representation to include semantic approximation in the process of querying databases, allowing the embedding and usage of large-scale unstructured and structured commonsense information into the querying process.

The proposed model can be applied to different data models such as RDF, relational, datalog and key-value stores. Despite its direct applicability as a generic semantic representation framework for structured data, the τ – Space framework fits into the broader *knowledge representation* (KR) discussion, targeting a KR framework with semantic approximation at its core.

This chapter is organized as follows. Section 6.2 analyzes the semantics of the data model behind RDF(S), which is the reference data model of this thesis. It also analyzes how different features of conceptual models for the RDF(S) can impact the ability to generate schema-agnostic queries (interpretability factors). Sections 6.3 and 6.4 formalizes the core representation elements of the τ – Space, which are described as an *inverted index structure* in Section 6.7. Section 6.8 analyzes the representation of dataset elements with complex predicates.

6.2 RDF(S) Data Model & Semantic Model

6.2.1 Motivation

In the *open communication scenario*, natural language descriptors associated with database elements play a fundamental role in the interpretation of the semantics of the database. A schema-agnostic query approach depends on the ability to automatically interpret the meaning of natural language descriptors. In this section we analyze the categories for lexical and link semantics associated with different elements in the RDF(S) data model, providing an initial framework to facilitate the integration between dataset descriptors and the distributional semantics representation.

6.2.2 Lexical Categories for the Data Model Elements

As databases are symbolic systems based on natural language, there is a natural isomorphism between *data model categories*, *logical types* and *lexical categories*. This correspondence is described below (grounded on the RDF(S) data model):

- **Instances:** Map to *named entities* which refer to the description of entities for which one or many rigid designators stands for the referent. Rigid designators include categories such as people, locations, events, biological species, substances, etc. A named entity is defined by one or more proper nouns (**NNP**) in a noun phrase (**NP**). Due to their specificity, named entities are less subject to vagueness, ambiguity and synonymy.

RDF(S) (EAV/CR)	Logic	Relational	Lexical Category
Instance	Constant	Value	NNP+
Value	Constant	Value	true—false—CD+
Class	Unary Predicate	Relation, Attribute	RB+—JJ+ NN(S)+ IN NNP+
Property	Unary, Binary Predicate	Entity, Attribute, Relation	BE VB IN, BE VB NN—JJ+

TABLE 6.1: Correspondence between RDF, Logics, Relational and lexical categories.

- **Classes:** Classes are *unary predicates* which map to *non-rigid designators*. *Non-rigid designators* are descriptors for sets of instances. Non-named entities (e.g. ‘*President of the United States*’) are more subject to vocabulary variation. Additionally, non-named entities have more complex compositional patterns: non-named entities can also be composed with named or non-named entities. A class is defined by one or more nouns (**NN**), adjectives (**JJ**), adverbs (**RB**), superlatives (**JJS**, **RBS**) in a noun phrase (**NP**) or adjectival phrase (**AP**) and can be connected to a named entity in a prepositional phrase (**PP**). Classes are commonly represented by sortal nouns [184] *apud* [185] [186].
- **Properties:** Properties are *binary predicates* which describe an *attribute, relationship, action* or *state*. It can be composed of nouns (**NN**), adjectives (**JJ**), adverbs (**RB**), comparatives (**RBR**, **JJR**), verbs (**VB**) in verbal phrases (**VP**). Prepositions play an important role in the definition of the directionality of the binary relation (e.g. ‘*child Of*’). Properties can be an *object properties*, where the range is defined by an instance or *data property*, where the range is defined by a value. Properties with boolean ranges functionally work as a unary predicate. Properties can refer to *relational nouns* [149] (‘brother’, ‘friend’, or meronymic relations). Relational nouns are not inherently unique.
- **Values:** Consists of numerical values in the real domain (cardinal & ordinal), dates, boolean values and strings.
- **Triple:** Consists in the representation of a fact with a <subject, predicate, object> structure. $\langle \{instance||class\} - \{property||type\} - \{instance||class||value\} \rangle$. Not all facts can be represented in one triple. On a normalized dataset scenario, a statement can be mapped to a conceptual model structure which entails multiple triples (e.g. as in the case of conceptual models for representing an event).
- **Context elements (reification):** A triple expressing a fact may depend on different contexts where the fact is embedded (such as a temporal, or spatial context). In a sentence this is expressed in a prepositional phrase (PP) or adverbial phrase (AdvP).

The RDF(S) data model categories have distinct linguistic patterns which can be used to define specific semantic approximation/interpretation approaches for different data model elements. The correspondence is depicted in Table 6.2.2.

6.2.3 Link Types

RDF datasets are labeled directed graphs in which the semantics of the edges can be classified according to the following high-level categories (Figure 6.1 Figure 6.1):

- **Class link:** Expresses a relationship between an instance and the unary predicate. The link connects unary predicates with a ‘*is a*’ semantics such as ‘place’, ‘person’, ‘plant’, ‘enzyme’, ‘anatomical structure’, ‘band’, ‘company’.
- **Relation link:** Expresses the *relationship* between two instances (e.g. ‘was born in’, ‘friend of’).
- **Attribute link:** Expresses the *attribute* of an instance (e.g. ‘age’, ‘gender’).
- **Co-reference link:** Expresses the *identity* or *equivalence relationship* between two instances, two properties or two classes (e.g. *owl:sameAs*, *rdfs:equivalentClass*, *rdfs:equivalentProperty*).
- **Description link:** A link which provides a *natural language descriptor* for the dataset element (e.g. *rdf:label*, *dcterms:description*, *rdfs:seeAlso*).
- **Context link:** Provides an attribute which expresses a context relation which is associated to an instance or in which a triple is valid. A context can be expressed using structural constructs, such as reification. Provenance descriptors are in this category.
- **Taxonomic link:** A *terminology-level* link which expresses a *specialization/generalization relation (taxonomical)* between *classes* or *properties* (e.g. *rdfs:subClassOf*, *rdfs:subPropertyOf*).
- **Structure link:** Corresponds to the links to *aggregation nodes* such as *blank nodes*.

RDF(S) provides a data model which supports major flexibility and variability in the way different database designers conceptualize datasets. Examples of link types from different datasets are shown in the listings below.

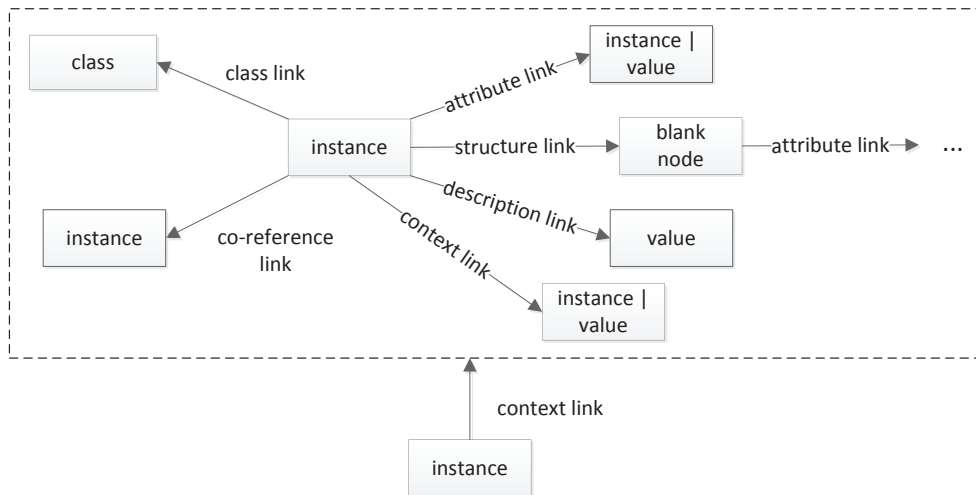


FIGURE 6.1: Link patterns at the instance-level.

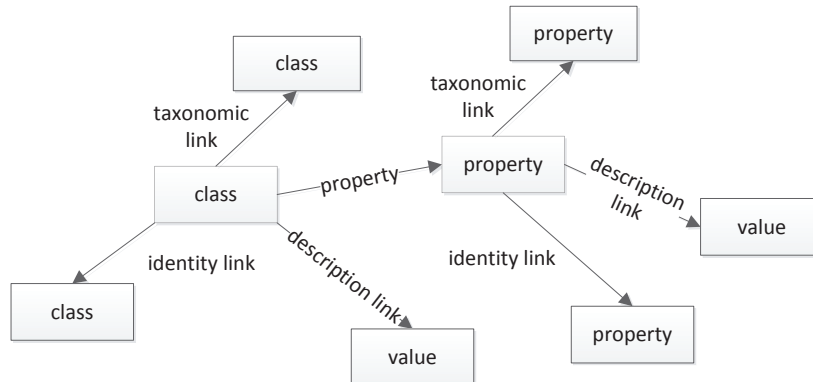


FIGURE 6.2: Link patterns at the terminology-level.

The identification of relation patterns (such as the ones expressed in the *Link Types* section) and their mapping to linguistic patterns, was already widely explored in the ontology learning literature, in particular by Voelker [187] and by Cimiano et al. [188]. The patterns identified in this section are a simplified subset of the patterns identified in the literature, which focus on the scenario of schema-agnostic query mapping.

Additionally, theories based on lexicon projection models [189] can be used to support a deeper and more principled analysis from the linguistic standpoint, where the emerging query/data predicate-argument structures could be explained by the empirical analysis of the lexical items, i.e. where syntactic phenomena could be ‘predicted’. This could provide a more sophisticated theory to explain and support query-data mappings with more complex compositional patterns. This dimension is not explored in the context of this work and it is left as future work.

```

yago:AfricanAmericanUnitedStatesSenators ,
yago:DemocraticPartyPresidentsOfTheUnitedStates ,
yago:AmericanCivilRightsLawyers ,
yago:HarvardLawSchoolAlumni ,
yago:OccidentalCollegeAlumni ,
yago:NobelPeacePrizeLaureates ,
yago:AmericanPoliticalWriters ,
yago:CommunityOrganizers ,
yago:IllinoisStateSenators ,
yago:UnitedStatesSenatorsFromIllinois

```

CODE 6.1: Triples with class link properties from the DBpedia/YAGO datasets.

```

dbpedia:Barack_Obama    dbpprop:hasPhotoCollection    ns221:Barack_Obama ;
dbpedia-owl:birthPlace dbpedia:Hawaii ,
    dbpedia:Honolulu ;
dbpprop:birthDate      "'1961-08-04'"^^xsd:date ;
dbpprop:birthName     "'Barack Hussein Obama II'"@en ;
dbpprop:birthPlace    "'Honolulu, Hawaii, U.S.'"@en ;
dbpprop:dateOfBirth   "'1961-08-04'"^^xsd:date ;
dbpprop:placeOfBirth  "'Honolulu, Hawaii, United States'"@en ;
dbpprop:website       <http://www.barackobama.com> ;
dbpprop:author        "'yes'"@en ;
dbpedia-owl:activeYearsEndDate "'2008-11-16'"^^xsd:date ,
    "'2004-11-04'"^^xsd:date ;
dbpprop:termStart     "'1997-01-08'"^^xsd:date ,
    "'2005-01-03'"^^xsd:date ,
    "'2009-01-20'"^^xsd:date ;
dbpedia-owl:residence dbpedia:White_House ;
dbpedia-owl:orderInOffice "'44th'"@en ;
dbpprop:office        "'President of the United States'"@en ,
    dbpedia:Illinois_Senate ,
    "'from the 13th District'"@en ;
dbpprop:party         dbpedia:Democratic_Party_(United_States) ;
dbpedia-owl:office    "'President of the United States'"@en ,
    "'Member of the Illinois Senate'"@en ,
    "'from the 13th District'"@en ;
dbpedia-owl:almaMater dbpedia:Harvard_Law_School ,
    dbpedia:Occidental_College ,
    dbpedia:Columbia_University ;
dbpprop:termEnd      "'2008-11-16'"^^xsd:date ,
    "'2004-11-04'"^^xsd:date ;
dbpprop:religion     dbpedia:Christian ;
dbpprop:residence    "'Chicago, Illinois'"@en ,
    dbpedia:White_House ;
dbpprop:profession   "'Author'"@en ,
    dbpedia:Community_organizing ,
    "'Constitutional law professor'"@en ,
    "'Lawyer'"@en ;
dbpprop:successor    dbpedia:Roland_Burris ,
    dbpedia:Kwame_Raoul ;
dbpprop:children     "'Sasha'"@en ,
    dbpedia:Family_of_Barack_Obama ;
dbpprop:spouse       dbpedia:Michelle_Obama ;
dbpedia-owl:activeYearsStartDate "'1997-01-08'"^^xsd:date ,

```

```

        ‘‘2005-01-03’’^^xsd:date ,
        ‘‘2009-01-20’’^^xsd:date ;
dbpedia-owl:profession  dbpedia:Community_organizing ,
dbpedia:Author ,
dbpedia:Constitutional_law ,
dbpedia:Lawyer ;
dbpedia-owl:religion    dbpedia:Christian ;
dbpedia-owl:party       dbpedia:Democratic_Party_(United_States) ;
dbpprop:almaMater       dbpedia:Occidental_College ,
        ‘‘Harvard Law School’’@en ,
        ‘‘Columbia University’’@en ;
dbpprop:predecessor     dbpedia:Peter_Fitzgerald_(politician) ,
dbpedia:Alice_Palmer_(politician) ,
dbpedia:George_W._Bush ;
dbpedia-owl:successor   dbpedia:Kwame_Raoul ,
dbpedia:Roland_Burris ;
dbpprop:order           44 ;
dbpedia-owl:child       dbpedia:Family_of_Barack_Obama ;
dbpedia-owl:region      dbpedia:Illinois ;
dbpedia-owl:seniority   ‘‘United States Senate’’@en ;
dbpedia-owl:spouse      dbpedia:Michelle_Obama .

```

CODE 6.2: Triples with attribute link properties.

```

<http://data.nytimes.com/47452218948077706853>
    owl:sameAs      dbpedia:Barack_Obama .

<http://lod.geospecies.org/ses/v6n7p>
dcterms:identifier  http://lod.geospecies.org/ses/v6n7p

```

CODE 6.3: Triples with co-reference link properties.

```

biolod:cria224u3ria224u1140i
    rdfs:label ‘‘decreased length in organ named hypocotyl
in environment of red light regimen for AT5G49230’’@en ;
    BiolOD_property_pria224u2i:annotation biolod:cria224u1ria224u683i .

dbpedia:Barack_Obama
    rdfs:label      ‘‘Barack Obama’’@en .
    dbpprop:shortDescription
        ‘‘American politician, 44th President of the United States’’@en ;

<http://lod.geospecies.org/ses/v6n7p>
    dcterms:title  Puma concolor (Linnaeus 1771)

default:AmazonOfferingABundle
    rdfs:comment ‘‘Amazon is offering a bundle, composed of s1234
phones and two batteries.’’^^xsd:string ;
    rdfs:seeAlso <http://www.amazon.com/cellphones/> ;

```

CODE 6.4: Triples with description link properties.

```

default:AmazonOfferingABundle
    gr:validFrom ‘‘2008-01-01T00:00:00Z’’^^xsd:dateTime ;

```

```
gr:validThrough '2008-12-31T23:59:59Z'^^xsd:dateTime .
```

CODE 6.5: Triples with context link properties.

```
eg:dataset-le1 a qb:DataSet;
  rdfs:label 'Life expectancy'@en;
  rdfs:comment 'Life expectancy within Welsh Unitary authorities
  - extracted from Stats Wales'@en;
  qb:structure eg:dsd-le ;
  sdmx-attribute:unitMeasure <http://dbpedia.org/resource/Year> ;
  .

eg:o1 a qb:Observation;
  qb:dataSet eg:dataset-le1 ;
  eg:refArea ex-geo:newport_00pr ;
  eg:refPeriod datagov:2004-01-01T00:00:00/P3Y ;
  sdmx-dimension:sex sdmx-code:sex-M ;
  eg:lifeExpectancy 76.7 ;
  .

eg:o2 a qb:Observation;
  qb:dataSet eg:dataset-le1 ;
  eg:refArea ex-geo:cardiff_00pt ;
  eg:refPeriod datagov:2004-01-01T00:00:00/P3Y ;
  sdmx-dimension:sex sdmx-code:sex-M ;
  eg:lifeExpectancy 78.7 ;
  .

eg:o3 a qb:Observation;
  qb:dataSet eg:dataset-le1 ;
  eg:refArea ex-geo:monmouthshire_00pp ;
  eg:refPeriod datagov:2004-01-01T00:00:00/P3Y ;
  sdmx-dimension:sex sdmx-code:sex-M ;
  eg:lifeExpectancy 76.6 ;
  .

...
```

CODE 6.6: Contextual relationship

Independently of domain, *link types* provide an abstract categorization across different conceptual models. The identification of link types help in the definition of the knowledge representation structure used in this work (τ – Space) and in the development of the schema-agnostic approach.

6.2.4 Factors Affecting Interpretability

A schema-agnostic approach needs to cope with differences in the way datasets are conceptualized. Despite the high-level link categories described in the previous section which correspond to the commonalities between different instances, there are feature

differences in the way datasets are built, which impact its interpretability from the perspective of a schema-agnostic query mechanism.

There are characteristics in the construction of the conceptual models and datasets which make them more or less friendly for schema-agnostic queries, i.e. which facilitate its interpretability for a third-party agent (automated or human), impacting the semantic matching between natural language and dataset. In this section we provide a *preliminary qualitative analysis* of these dimensions based on the manual analysis of the set of 15 open datasets: DBpedia¹, YAGO², New York Times³, DrugBank⁴, Bio2RDF⁵, MusicBrainz⁶, BioLOD⁷, Geospecies⁸, CIA World FactBook⁹, Vulnerapedia¹⁰, different dataset instances of the Good Relations vocabulary¹¹, Data.gov.uk (Patents, Crime)¹², OpenCorporates¹³, Citeseer¹⁴ and LinkedCT¹⁵, SIFEM Inner Ear Data¹⁶, XBRL dataset¹⁷.

Five major interpretability factors were identified:

- *Structural Granularity*: Labels associated with entities can vary in their size (number of words used to describe it). While some descriptors can be very simple such as ‘Bill Clinton’ (instance) or ‘spouse’ (property), others can have a very complex word composition, such as ‘Democratic Party Presidents Of The United States’ (DBpedia class) and ‘Derivative Instruments Gain Loss Recognized In Income Ineffective Portion And Amount Excluded From Effectiveness Testing Net’ (us-gaap class), or ‘decreased length in organ named hypocotyl in environment of red light regimen for AT5G49230’ (instance). The complexity in the individual descriptor associated with the entity has an intrinsic relation to the level of structure used in the description of the entities in the dataset. While some datasets may compose concepts through structural relations, others define complex concepts by having large natural language descriptors associated with the entity. More structured representations are more expressive, supporting a larger number of queries over the data as they support a larger number of combinations of structural

¹<http://dbpedia.org/>

²<http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>

³<http://data.nytimes.com/>

⁴<http://datahub.io/dataset/fu-berlin-drugbank>

⁵<http://bio2rdf.org/>

⁶<https://musicbrainz.org/>

⁷<http://biolod.jp/about>

⁸<http://datahub.io/dataset/geospecies>

⁹<http://datahub.io/dataset/cia-world-factbook>

¹⁰<http://datahub.io/dataset/vulnerapedia>

¹¹<http://www.heppnetz.de/projects/goodrelations/>

¹²<http://data.gov.uk/>

¹³<https://opencorporates.com/>

¹⁴<http://citeseer.rkbexplorer.com/sparql/>

¹⁵<http://linkedct.org/>

¹⁶<http://www.sifem-project.eu/>

¹⁷<http://datahub.io/dataset/semantic-xbrl>

relations. Additionally, more structured representations tend to have a positive impact for supporting schema-agnostic queries, as primitive concepts are explicitly represented on the schema and vocabularies tend to be more centralised and normalised.

- *Generality/Specialisation Level:* More specialised domains tend to have higher specificity and concepts tend to be expressed through more complex compositional patterns (higher structural complexity) (for example the SIFEM Inner-ear Dataset¹⁸ and the RDF version of the XBRL dataset¹⁹). The complex compositional patterns increases the probability of a matching error (less supportive for schema-agnostic queries). On the other hand, very specialized domains tend to have less variable terminologies, being less bound to ambiguity, vagueness and synonymy conditions. In contrast, more generic/open domains are more bound to vagueness, ambiguity and synonymy but tend to have lower structural complexity.
- *Completeness Level:* A conceptual model is an artefact which expresses an explicit formal conceptualization needed to address a specific task. The level of completeness of the conceptual model is relative to the reference frame of the task in which the conceptual model is designed to address. However, the conceptual models behind the datasets can be designed to target a more complete description of a domain. More complete descriptions increase the probability of an exact semantic matching between the query terms and dataset terms.
- *Presence of Extra-linguistic Features:* Some domains are dependent on elements outside the domain of linguistic expression. Models in Physics, Engineering or Mathematics (Such as the SIFEM conceptual model [190]) might depend on the representation of objects which depend on geometrical, topological or abstract descriptive features. It is still possible to target schema-agnostic queries in this context, however, other query forms that are less dependent on linguistic information (visualisations and other visual interaction elements) may need to be employed. Coping with this type of conceptual model is outside the scope of this work.
- *Structural Complexity:* Datasets with complex relational patterns will resort to implicitly defined structures to represent complex associations. These datasets will become more distant from the natural language syntax in the direction of an artificial structure, which reflects complex structural relations in the described domain (e.g. meronymic relations) where *blank nodes* and arbitrary identifiers are used to structure the information. The description of parts in a complex mechanism (e.g. anatomical relations and physiological processes).

¹⁸<http://www.sifem-project.eu/>

¹⁹<http://datahub.io/dataset/semantic-xbrl>

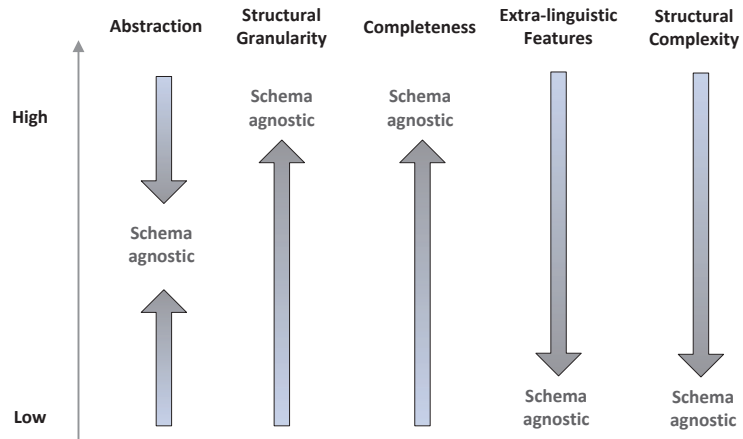


FIGURE 6.3: Analysis of the interpretability dimensions of databases.

Figure 6.3 depicts the five *interpretability* elements and the regions in which interpretability is maximized.

6.3 Distributional-Relational Model (DRM)

As mentioned in Section 4.2.3 the semantics of a database element e is represented by the set of descriptors associated with e . This typically does not include concept associations which are outside the scope of the specific task that the database is aiming to address, which limits its use for purposes of semantic approximation to concepts outside the specific database design context. Semantic approximation operations based on semantic/commonsense knowledge are fundamental for semantic (schema-agnostic) queries, or to database integration where users are not aware of the representation of the database.

In the τ – Space representation model, the formal semantics of a database symbol is extended with its distributional semantics description, which captures the large-scale semantic/commonsense associations under a reference corpora. The distributional semantics representation captures the large-scale *semantic*, *commonsense* and *domain specific knowledge*, using it in the *semantic approximation* process between a user information need and the database. The *hybrid distributional-structured model* is called *Distributional-Relational Model* (DRM).

A DRM embeds the structure defined by structured data models in a distributional vector space, where every labeled entity has an associated distributional vector representation. The distributional associational information embedded in the distributional

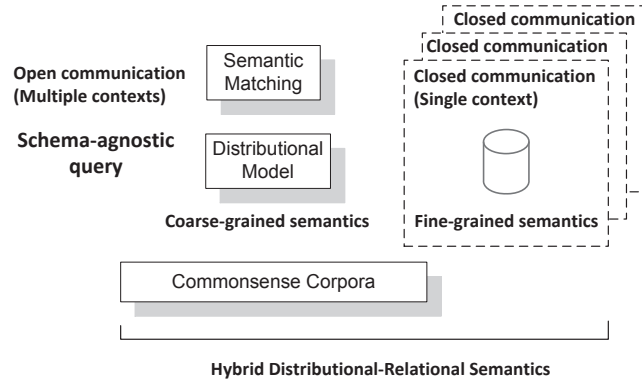


FIGURE 6.4: Distributional semantics layer complementing the database semantics.

vector space is used to semantically complement the knowledge expressed in the structured data model. The distributional information is then used to support semantic approximations, while preserving the structured data semantics.

Definition 6.1 (Distributional-Relational Model (DRM)). A *Distributional-Relational Model (DRM)* is a tuple $(DSM, DB, RC, \mathcal{F}, \mathcal{H})$, where: DSM is the associated *distributional semantic model*; DB is the *database* with elements E ; RC is the *reference corpora* which can be unstructured, structured or both. The reference corpora can be internal (based on the co-occurrence of elements within the DB) or external (a separate reference corpora); \mathcal{F} is a *map* which translates the elements $e_i \in E$ into vectors \vec{e}_i in the the distributional vector space VS^{DSM} using the natural language descriptor of e_i ; and \mathcal{H} is the set of thresholds above which two terms are semantically equivalent.

- DSM is the *associated distributional semantic model*.
- DB is the *structured dataset* with DB elements E and tuples T .
- RC is the *reference corpora* which can be unstructured, structured or both. The reference corpora can be internal (based on the co-occurrence of elements within the DB) or external (a separate reference corpora).
- \mathcal{F} is a *map* which translates the elements $e_i \in E$ into vectors \vec{e}_i in the the distributional vector space VS^{DSM} using the string of e_i and the data model category of e_i .
- \mathcal{H} is the set of *semantic thresholds* for the distributional semantic relatedness s in which two terms are considered semantically equivalent if they are equal and above the threshold.

Definition 6.2 (Distributional interpretation of a database element). The distributional interpretation of a database element e under a DSM and reference corpora RC is represented as $[[\vec{e}]]^{DSM(RC)}$.

6.4 Distributional Structured Semantic Space (τ – Space)

6.4.1 Introduction

The main elements of the DRM are realized into a vector space model named τ – Space, which defines a *distributional structured vector space model* (VSM).

The distributional semantic space is composed by a segmented vector space where the vector space is segmented into *subspaces* associated with data model categories (*instances*, *properties* and *classes*). The segmentation and the relationship between subspaces reflects the graph structure of the data in the *DB*. The *space segmentation* also works as a local and contextualised *dimensional reduction technique*.

The construction strategy for the τ – Space supports the creation of different semantic approximation criteria for distinct data model categories. The different approximation approaches are reflected both in the definition of the vector space basis and in the distance measures. The rationale behind this approach is that different data model categories demand different semantic representation and approximation approaches.

The process of semantic matching consists in projecting the query terms into the τ – Space and computing the semantic distance in relation to the database elements, respecting the *syntactic constraints* of the query and of the database triples. This chapter focuses on the description of the distributional semantics representation framework for the *DB* data, while Chapter 8 targets the description of the query process.

6.4.2 Building the τ – Space

The construction of the τ – Space takes into account the different representational and semantic approximation demands of different data model categories.

From the semantic approximation perspective, each data model category has the following requirements:

- **Instances:** In a typical *DB*, instances are the most numerous data model category. Because instances are less bound to vocabulary variation, in most of the cases no semantic approximation is necessary. In some cases the approximation consists of substring approximation operations (*Clinton* \rightarrow *Bill Clinton*). In more rare cases aliasing can be observed, such as in *Lawrence of Arabia* \rightarrow *T.E. Lawrence* or *Edson Nascimento* \rightarrow *Pele*. In these cases, distributional semantic approximations can be applied to address aliasing.

- **Properties:** Properties are bound to a high vocabulary variation. In most of the cases, the property consists of a *content word* (noun, adjective and verb) which can be complemented by a preposition, auxiliary verb (e.g. is, has), or less frequently, an adverb. The semantic representation model for this category should be able to address vocabulary variation and ambiguity and express the directionality of the predication. In more rare cases, complex compositional patterns can be observed.
- **Class:** The semantic representation for classes should be able to cope with high vocabulary variation. Classes have complex compositional patterns and can commonly have more than one content word (see Section 6.8): for complex class descriptors, the semantic matching will depend on a principled compositional approach. In the context of this work we classify the classes into two types: (i) *class* (≤ 2 words), (ii) *complex class* ($>$ three words).

The construction of the τ – Space uses a distributional semantic approximation for properties and classes and primarily a string approximation for instances (complemented by a distributional approach to detect aliasing). In the next sections the geometric representation of the τ – Space model is defined.

6.4.3 Vector Basis

The τ – Space is built from three main types of dimensional bases:

1. *Distributional Reference Frame* (VS^{dist}): Consists of vector space dimensions which represent weighted context vectors over a reference corpora RC . The distributional reference frame is defined by taking all terms associated with elements from the data model category which has a distributional representation, getting their context vectors and associated weights. The distributional context vector for a term t is called the distributional interpretation of t ($[[t]]_{dist}$) and can be built in two ways: (i) by defining a maximum vector length or (ii) by getting all the contexts to which t is associated. VS^{dist} is the vector space spanned by the distributional basis vectors defined for all $e \in DB$. Vectors weights are defined in the $[0,1]$ interval.
2. *Word Reference Frame* (VS^{word}): Consists of a reference frame in which each dimension represents a word occurring in the terms associated with database elements E . A vector for a term t is defined as a weighted vector of words, where the vector weights are a function of the frequency of occurrence of w in DB . VS^{word} is the vector space spanned by the word basis vectors. Word weights are defined in the $[0,1]$ interval.

3. *Ordered dimension*: R : Consists of the real dimension for the representation of numerical entities in the database.

6.4.4 Word Space (VS^{word})

Definition 6.3 (Word vector basis). Let E be the set of entities in the database DB . Let T be the set of target terms associated with the entities E . Each $t \in T$ is composed of a set of k words. Let K be the set of all words in the database lexicon. Let $w_{i,j} > 0$ be a weight associated with each word k_i contained in a database tuple d_j , where for a k_i word not contained in a tuple d_j , $w_{i,j} = 0$. The set of words $K = \{k_1, \dots, k_t\}$, of all words available in \mathbb{DB} is used to define the basis $Word_{base} = \{\vec{\mathbf{k}}_1, \dots, \vec{\mathbf{k}}_t\}$ of unit vectors that spans the *word vector space* VS^{word} .

The set of k_i words defines a unitary coordinate basis for the vector space. Representing the target term in relation to the set of basis word vectors:

$$\mathbf{t} = \sum_{i=1}^t w_{i,j} \mathbf{k}_i, (j = 1, \dots, N) \quad (6.1)$$

Different weighting schemes based on frequencies can be applied to the word vector. In this work the TF/IDF weighting scheme [191] is used.

Definition 6.4 (Term Frequency). Let $freq_{i,j}$ be the frequency of term k_i in the tuple \mathbf{d}_j . Let $count(\mathbf{d}_j)$ be the number of words inside the tuple \mathbf{d}_j . The normalized term frequency $tf_{i,j}$ is given by:

$$tf_{i,j} = \frac{freq_{i,j}}{count(\mathbf{d}_j)} \quad (6.2)$$

Definition 6.5 (Inverse Document Frequency). Let n_{k_i} be the number of tuples containing the word k_i and N the total number of tuples. The *inverse document frequency* for the word k_i is given by:

$$idf_i = \log \frac{N}{n_{k_i}} \quad (6.3)$$

Definition 6.6 (Term Frequency/Inverse Document Frequency). The final TF/IDF weight value based on the values of tf and idf is defined as:

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{n_{k_i}} \quad (6.4)$$

where the weight given by TF/IDF provides a measure on how a term is discriminative in relation to the relative distribution of other terms in the DB .

6.4.5 Distributional Space (VS^{dist})

Definition 6.7 (Context vector basis). Let $Context = \{c_1, \dots, c_t\}$ be the set of distributional contexts patterns which are defined by the target terms associated with a reference corpus RC . This set is used to construct the basis $Context_{base} = \{\vec{\chi}_1, \dots, \vec{\chi}_t\}$ of vectors that spans the *distributional vector space* VS^{dist} .

The interpretation of a term t in relation to the distributional corpus is given by:

$$\vec{t} = \sum_{i=1}^t v_i^t \vec{c}_i \quad (6.5)$$

Using the *Einstein summation convention*, a document has its associated concept vector:

$$\mathbf{e} = V^i \chi_i \quad (6.6)$$

This work uses the concept of the *generalized vector space model* (GVSM) introduced in [192] and [193] in the context of DSMs. In the GVSM model, Wong et al. [194] propose an interpretation of the term vectors present in the index as linearly independent but not pairwise orthogonal.

In the term VSM, the term vectors have unit length and are orthogonal. Embedded in these conditions is the assumption that there is no interdependency between terms (non-correlated terms) in the corpus which defines the document collection [194]. The generalized vector space model (GVSM) takes into account *term interdependency*, generalizing the identity matrix which represents $\vec{\mathbf{k}}_i \cdot \vec{\mathbf{k}}_l$ into a matrix G with elements $g_{i,l}$.

6.4.6 Unification of the Distributional and the Word Spaces

Both distributional and word spaces can be unified in a single space, where the dimensions of the word space can be defined using coordinates of the distributional space. This

allows the transformation of a term vector in the word space VS^{word} into a distributional space representation VS^{dist} and vice-versa. A vector $\vec{x} \in VS^{dist}$ can be mapped to VS^{word} by the application of the following transformation:

$$\vec{x} = \sum_{i=1}^t \alpha_i v_i^x \vec{k}_i \quad (6.7)$$

where α_i is a second-order transformation tensor which is defined by the set of word vectors of distributional contexts.

6.4.7 Instance subspaces (VS^I)

Let I be the set of instances in the database DB , where every instance $i \in I$ have an associated natural language identifier. Each i can be defined as a vector over VS^{word} and VS^{dist} :

$$\vec{\mathbf{I}}_{VS^{word}} = \{ \vec{\mathbf{i}} : \vec{\mathbf{I}} = \sum_{j=1}^t w_j^i \vec{k}_j, \text{ for each } i \in I \} \quad (6.8)$$

$$\vec{\mathbf{I}}_{VS^{dist}} = \{ \vec{\mathbf{i}} : \vec{\mathbf{I}} = \sum_{j=1}^t v_j^i \vec{\chi}_j, \text{ for each } i \in I \} \quad (6.9)$$

The potentially large number of instances can define distributional spaces with very high dimensionality, impacting the computation of vector distance operations. Since instances have lower vocabulary variability, VS^{word} can be used as the primary query space for instance queries, using the VS^{dist} as a secondary search space for searching for instance aliases.

6.4.8 Property subspaces (VS^P)

Due to their high vocabulary variation, properties are defined over the distributional reference frame VS^{dist} . Two approaches are used for the distributional representation of properties:

Non-contextualised: Where the distributional vector space is built by adding the distributional vector of all the properties to the space:

$$\vec{\mathbf{P}}_{VS^{dist}} = \{ \vec{\mathbf{p}} : \vec{\mathbf{P}} = \sum_{i=1}^t v_i^p \vec{\chi}_i, \text{ for each } p \in P \} \quad (6.10)$$

Contextualised: In which the vector space associated with the property is defined under the context of an instance i in which the property occurs. This defines a vector space $VS^P(i)$ which is parametrised by the instance i .

$$\overrightarrow{\mathbf{P}(i)}_{VS^{dist}} = \{\overrightarrow{\mathbf{p}(i)} : \overrightarrow{\mathbf{P}(i)} = \sum_{j=1}^t v_j^p \overrightarrow{\chi}_j, \text{ for each } p_j(i, x) \vee p_j(x, i) \in DB\} \quad (6.11)$$

The contextualised method has two advantages in relation to the non-contextualised method: (i) it improves the accuracy of the semantic matching, reducing the influence of properties which are not associated with the context of a specific semantic matching (*semantic pivot*); (ii) it reduces the dimensionality of the space, representing only the subspaces of properties associated with a specific instance, improving the temporal performance of the computation of similarity measures.

A query over the non-contextualised vector space is used when a property is a semantic pivot or when the query depends on the look-up of all properties in the DB .

The geometric vectors $\overrightarrow{\mathbf{p}}$ and $\overrightarrow{\mathbf{p}(i)}$ are identical, differing just in the vector space which is spanned by them. The directional and syntactic component of properties are defined in section 6.4.11.

6.4.9 Class subspaces (VS^C)

Classes can have both high vocabulary variability and also complex compositional patterns. This section concentrates on a simplified description of the class representation, focusing on regular classes (described by ≤ 2 words), while section 6.8 focuses on complex classes (> 2 words).

Similarly to properties, classes can be interpreted in the context of an instance $VS^C(i)$ or for all the classes VS^C :

Non-contextualised: Where the distributional vector space is built by the distributional vector of all the classes:

$$\overrightarrow{\mathbf{C}}_{VS^{dist}} = \{\overrightarrow{\mathbf{c}} : \overrightarrow{\mathbf{C}} = \sum_{j=1}^t v_j^c \overrightarrow{\chi}_j, \text{ for each } c \in C\} \quad (6.12)$$

Contextualised: In which the vector space is associated with the context of an instance i to which the class is associated.

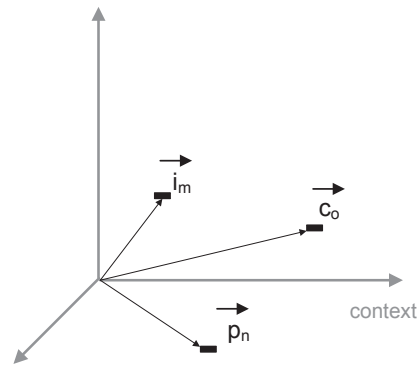


FIGURE 6.5: Distributional vector representation for instances, properties and classes.

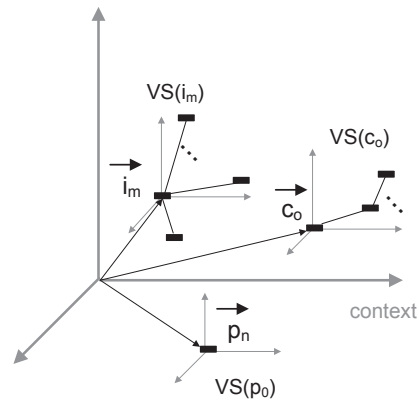


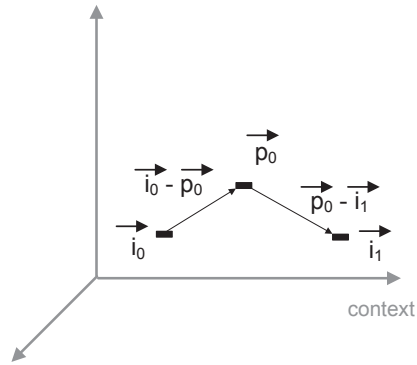
FIGURE 6.6: Vector representation for the distributional subspaces associated with instances (contextualised).

$$\overrightarrow{\mathbf{C}(\mathbf{i})}_{VS^{dist}} = \{\overrightarrow{\mathbf{c}(\mathbf{i})} : \overrightarrow{\mathbf{C}(\mathbf{i})} = \sum_{j=1}^t v_j^c \overrightarrow{\chi}_j, \text{ for each } c_j(i) \in DB\} \quad (6.13)$$

Figure 6.5 and Figure 6.6 depict the non-contextualised and the contextualised subspaces.

6.4.10 Real dimension (\mathbb{R})

Expresses numeric values in the real domain \mathbb{R} in a one-dimensional subspace. Maps to numerical value data types.

FIGURE 6.7: Property relation vectors in the τ – Space.

6.4.11 Property Relation Vectors

With the definition of vectors for every data model category, a vector representation for the syntactic relations between properties and instances can be defined. The vector representation of a triple $r = p(i_1, i_2)$ in the contextualised vector space is defined by:

Definition 6.8 (Distributional Representation of a Property Assignment Triple). Let \vec{p} , \vec{i}_1 and \vec{i}_2 be the vector representation of the binary predicate elements p, i_1 and i_2 in $p(i_1, i_2)$. A triple vector representation (denoted by \vec{r}) is defined by: $(\vec{p} - \vec{i}_1, \vec{i}_2 - \vec{p})$ if $p(i_1, i_2)$.

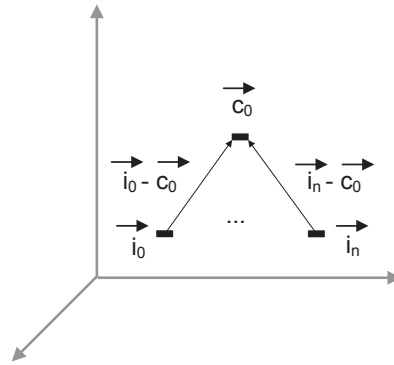
Relation vectors are defined over the $VS^{dist} = VS^I \cup VS^P$. Figure 6.7 depicts the vector representation of a property triple.

6.4.12 Class Relation Vectors

For the relation between instance and class, the vector representation of a triple $r = c(i)$ in the vector space is given by the following definition:

Definition 6.9 (Distributional Representation of a Class Assignment Triple). Let \vec{c} , \vec{i} be the vector representations, of the unary predicate c and the instance i . A triple vector representation (denoted by \vec{r}) is defined by: $(\vec{c} - \vec{i})$.

Figure 6.7 depicts the vector representation of a class triple. The class assignments are defined over the $VS^{dist} = VS^I \cup VS^C$.

FIGURE 6.8: Class relation vectors in the τ - Space.

6.4.13 Building the τ - Space

The construction of the τ - Space depends on the alignment between the different subspaces. Depending on the motivation for its construction, there are two possible approaches for the construction of the τ - Space:

- **Single-Space Model:** Unifies all subspaces into a single vector space, targeting the maximization of the geometric interpretation of the τ - Space. The main disadvantage of this approach is the growth in the dimensionality of the unified space.
- **Multiple-Spaces Model:** τ - Space as a composition of interrelated vector spaces. While this approach does not have a simple geometrical/topological mathematical interpretation under the context of vector analysis, it provides a representation which facilitates the local reduction of the dimensionality of the vector space, by segmenting the space into different vector spaces. Due to its practical implications, this work concentrates on the multiple-spaces model.

The structure of the τ - Space is defined by the relationship between the different subspaces (Figure 6.9), which is defined by the RDF(S) data model syntax/grammar. In Figure 6.9, the boxes represent the subspaces for different data model elements (class, instance, property, value), while the arrows represent the syntactic relationships between these elements, where specific RDF(S) vocabularies were omitted. Elements such as $\langle P, P \rangle$, $\langle C, C \rangle$ for example, describe the taxonomical relation between properties and classes respectively, while $\langle P, I \rangle$ $\langle I, P \rangle$ are property relations and $\langle C, I \rangle$ $\langle I, C \rangle$ are class relations.

The contextualised subspaces $VS^P(i)$ and $VS^C(i)$ are vector spaces which are parametrized by an instance i , while the VS^P and VS^C are subspaces which contain all properties

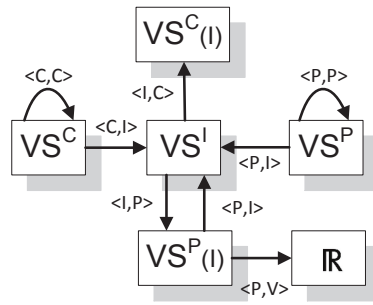


FIGURE 6.9: Topological relationship between the vector spaces that generate the τ – Space.

and classes from the *DB*. The relations between different vector spaces are depicted as arrows in Figure 6.9.

The procedure for the construction of the τ – Space under the multiple vector spaces model is described in Algorithm 29.

The topology of the multiple-spaces τ – Space does not have a trivial geometrical/topological interpretation, defining a structure which is similar to a *fibration*. A *fibration* is a generalization of the notion of a *fiber bundle* [195]. The fibration is a mathematical model structure which models a topological space being parameterized by another topological space (which is called a *base*) [196]. In the τ – Space the base is defined by the instances, which parametrizes the property and class vector spaces.

6.4.14 τ – Space Example

Figures 6.10, 6.11, 6.12, 6.13 and 6.14 depict the steps for the construction of the τ – Space for the example dataset:

DB:

```
:children(:Bill_Clinton, :Chelsea_Clinton)
:spouse(:Chelsea_Clinton, :Marc_Mezvinsky)
:PresidentsOfTheUnitedStates(:Bill_Clinton)
...
```

The τ – Space construction starts with the definition of term and distributional vectors for instances (:Bill_Clinton, :Chelsea_Clinton, :Marc_Mezvinsky, ...), defining VS^I (VS^{word}, VS^{dist}), Figures 6.10. The next step defines the vector space for all classes VS^C (VS^{word}, VS^{dist}) (Figure 6.11) and properties VS^P (VS^{dist}) (Figure 6.12). The

Algorithm 1 τ – Space Construction**INPUT**

- DB : The RDF(S) database with a data model $\Sigma = (I, P, C, V)$.
- $DSM(RC)$: The distributional semantic model over the corpora RC .

OUTPUT

- τ – Space: .

PROCEDURE *BuildContextualisedT-Space()*:

```

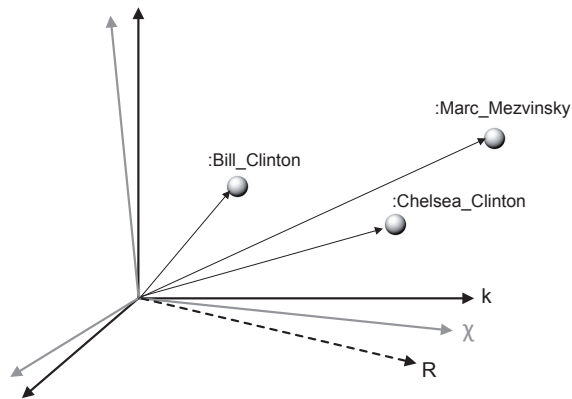
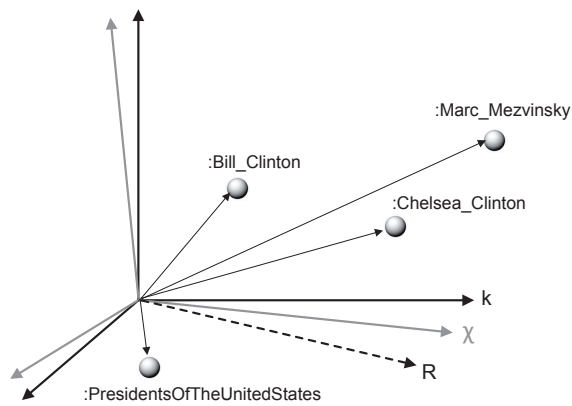
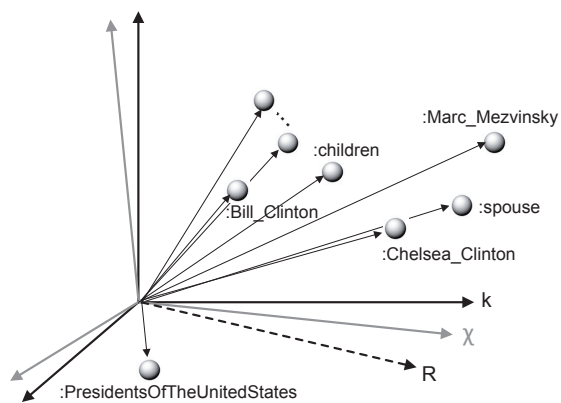
1: for all  $i \in I$  do
2:    $\vec{k} \leftarrow \text{getWordVector}(i)$ 
3:    $\text{addToVectorSpace}(VS^{I(\text{term})}, \vec{k})$ 
4:    $\vec{i} \leftarrow \text{getDistributionalVector}(i, DSM(RC))$ 
5:    $\text{addToVectorSpace}(VS^{I(\text{dist})}, \vec{i})$ 
6:    $\text{createSubspace}(VS^{P(\text{dist})}(i))$ 
7:   for all  $p \in p(i, x)$  do
8:      $\vec{p} \leftarrow \text{getDistributionalVector}(p, DSM(RC))$ 
9:      $\text{addToVectorSpace}(VS^{P(\text{dist})}(i), \vec{p})$ 
10:     $\text{addToVectorSpace}(VS^{P(\text{dist})}, \vec{p})$ 
11:     $\vec{r} \leftarrow \vec{i} - \vec{p}$ 
12:     $\text{addToVectorSpace}(VS^{R(\text{dist})}, \vec{r})$ 
13:   end for
14:   for all  $p \in p(x, i)$  do
15:      $\vec{p} \leftarrow \text{getDistributionalVector}(p, DSM(RC))$ 
16:      $\text{addToVectorSpace}(VS^{P(\text{dist})}(i), \vec{p})$ 
17:      $\text{addToVectorSpace}(VS^{P(\text{dist})}, \vec{p})$ 
18:      $\vec{r} \leftarrow \vec{p} - \vec{i}$ 
19:      $\text{addToVectorSpace}(VS^{R(\text{dist})}, \vec{r})$ 
20:   end for
21:   for all  $c \in c(i)$  do
22:      $\vec{c} \leftarrow \text{getDistributionalVector}(c, DSM(RC))$ 
23:      $\text{addToVectorSpace}(VS^{C(\text{dist})}(i), \vec{c})$ 
24:      $\text{addToVectorSpace}(VS^{C(\text{dist})}, \vec{c})$ 
25:      $\vec{r} \leftarrow \vec{i} - \vec{c}$ 
26:      $\text{addToVectorSpace}(VS^{R(\text{dist})}, \vec{r})$ 
27:   end for
28: end for
29: return  $VS^{P(\text{dist})}, VS^{C(\text{dist})}, VS^{I(\text{dist})}, VS^{R(\text{dist})}$ 

```

third step consists in the creation of the parametrized distributional vector spaces for properties and classes $VS^P(i)$, $VS^C(i)$ (VS^{dist}) (Figure 6.11, 6.12, 6.13). In the last step the triple vectors are computed (Figure 6.14).

6.4.15 Complementary Structures: Reification subspace

Reification can be used as a mechanism to consistently represent contextual information in a schema-agnostic way. The reification subspace is defined as a vector space which is

FIGURE 6.10: Creation of the instance subspaces VS^I (VS^{word} , VS^{dist}).FIGURE 6.11: Creation of the class subspaces VS^C (VS^{word} , VS^{dist}).FIGURE 6.12: Creation of the property subspaces VS^P (VS^{dist}).

parametrized by a triple d , where $d \in D$ and $D = I \times C \times P \times V$. A triple d with a set of contexts Ω of the form $\langle d, \omega, \{i||v\} \rangle$.

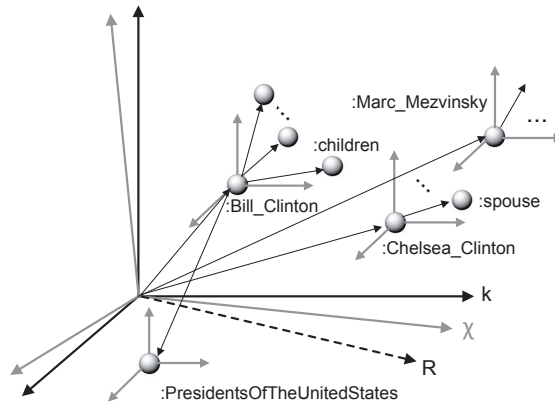


FIGURE 6.13: Creation of the parametrized subspaces $VS^P(i)$, $VS^C(i)$ (VS^{dist}).

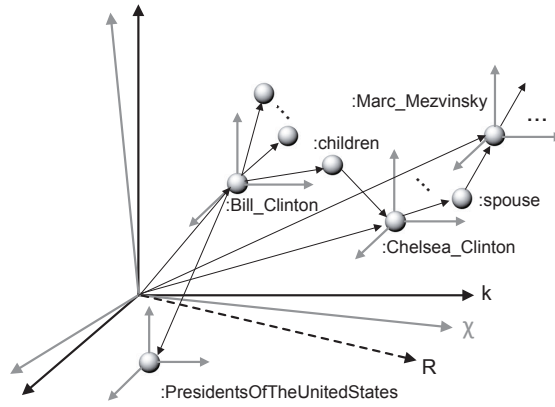


FIGURE 6.14: Creation of the relation vectors.

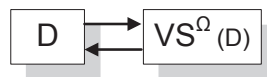


FIGURE 6.15: Reification extension of the core τ – Space.

$$\overrightarrow{\Omega(\mathbf{d})}_{VS^{dist}} = \{\overrightarrow{\omega(\mathbf{d})} : \overrightarrow{\Omega(\mathbf{d})} = \sum_{i=1}^t v_i^\omega \overrightarrow{\chi}_d, \text{ for each } \omega_j(d, x) \vee \omega_j(x, d) \in DB\} \quad (6.14)$$

Figure 6.15 depicts the τ – Space representation for the reification.

6.4.16 Compositionality & Semantic Interpretation

Compositional-distributional models typically represent a sentence or expression as a single interpretation vector. Most compositional-distributional models define the projection of syntactic relations as vector transformation operations in the vector space (using additive or multiplicative models) [155]. The process of interpreting a sentence consists in the application of successive vector transformation operations according to the grammar [155].

This work uses a different strategy to represent the interpretation vector of a triple. Instead of collapsing the three triple elements $\langle s, p, o \rangle$ into a single vector, the interpretation of the triple is set of syntactically connected vectors in the τ – Space.

Definition 6.10 (Distributional Interpretation of a Triple). The *distributional interpretation* of a triple d is given by:

- the vector tuple $\langle \vec{i}_1, \overrightarrow{\mathbf{p}(\mathbf{i}_1, \mathbf{i}_2)}, \{\vec{i}_2 \| v\}, (\vec{\mathbf{p}} - \vec{i}_1), (\vec{i}_2 - \vec{\mathbf{p}}) \rangle$ where $\vec{i}_1 \in \vec{\mathbf{I}}$, $\vec{\mathbf{p}} \in \vec{\mathbf{P}}$ and $v \in V$, for the triple of the form $\langle i, p, \{i, v\} \rangle$.
- the vector tuple $\langle \vec{i}, \overrightarrow{\mathbf{c}(\mathbf{i})}, (\vec{\mathbf{c}} - \vec{i}) \rangle$ where $\vec{i} \in \vec{\mathbf{I}}$, $\vec{\mathbf{c}} \in \vec{\mathbf{C}}$ and $v \in V$, for the triple of the form $\langle I, type, C \rangle$

6.5 Discussion

The proposed approach introduced in this work embeds an RDF graph into a vector space, adding a geometrical interpretation for the data elements. The vector space is built from a distributional model, where the coordinate reference frame is defined by interpretation vectors mapping the statistical context distribution of terms in the reference corpora. This distributional coordinate system supports a semantic representation of the RDF graph elements which allows a flexible semantic search and matching between query terms and database elements.

Distinctive characteristics of the τ – Space and distributional semantic approaches applied in Information Retrieval (IR) such as *Latent Semantic Indexing* (LSI) [170] are:

- **External distributional knowledge source:** The use of an *external distributional data source* which targets a commonsense and semantic knowledge base, instead of using the information from the indexed dataset, providing a more comprehensive distributional reference frame.

- **Preservation of the structural/syntactic information of the dataset:** Each labeled entity in the DB defines a point in the τ – Space. The topological structure of the data graph (set of database tuples) maps to the set of relation vectors. The set of relation vectors associated with each entity point, defines another difference in relation to traditional VSMs and distributional models for IR. Comparatively, traditional VSMs represent documents as (free) vectors at the origin of the vector space and remove the syntactic relation information, collapsing it into a bag of words.
- **Use of distributional semantics to represent database entity semantics:** Most IR approaches using distributional semantics have focused on defining distributional vector space models for indexing the content of unstructured documents, using a bag-of-words approach, where no fine-grained compositional model is defined for the document sentences. The use of a distributional vector for the representation of entity semantics in a database scenario is also a distinctive characteristic of the proposed model, which allows the distributional semantics approximation in relation to short natural language expressions (database descriptors for the entities in the dataset), where the impact on the lack of a compositional model is reduced, maximizing the effectiveness of the distributional semantic approximation.

6.5.1 Transportability as Coordinate Transformations

Vector and tensor quantities can be represented in relation to a reference frame (coordinate system). Under this representation, however, changes of reference frame imply a change in the representation of the object. However, the transformation rules associated with the changes of reference frame are well defined objects and the representation of the object in a different reference frame can be recalculated. Tensors can be seen as geometric objects represented by numeric arrays that transform according to certain rules under a change of coordinates.

The capacity to transform objects in the τ – Space across different coordinate systems can support the transportability across different distributional models. Data graphs from different domains can be supported by different distributional models, instead of a *one size fits all* solution. While an open domain data graph like DBpedia can be supported by a distributional model derived from Wikipedia, a domain specific data graph covering financial data can use a distributional model built from a domain specific financial reference corpus. Spaces with different distributional models can form patches in a more complex distributional manifold. Additionally, different distributional models can be used in parallel to support multiple interpretations of the elements embedded in the space.



FIGURE 6.16: Dimensions of the distributional tensor.

6.6 Tensor Representation

The τ – Space can also be represented using a tensor representation. The core distributional tensor has four main dimensions, three corresponding to the representation of the triple data and one corresponding to the distributional vector for each triple element $R_{spo\chi}$.

Assuming the following example *DB*:

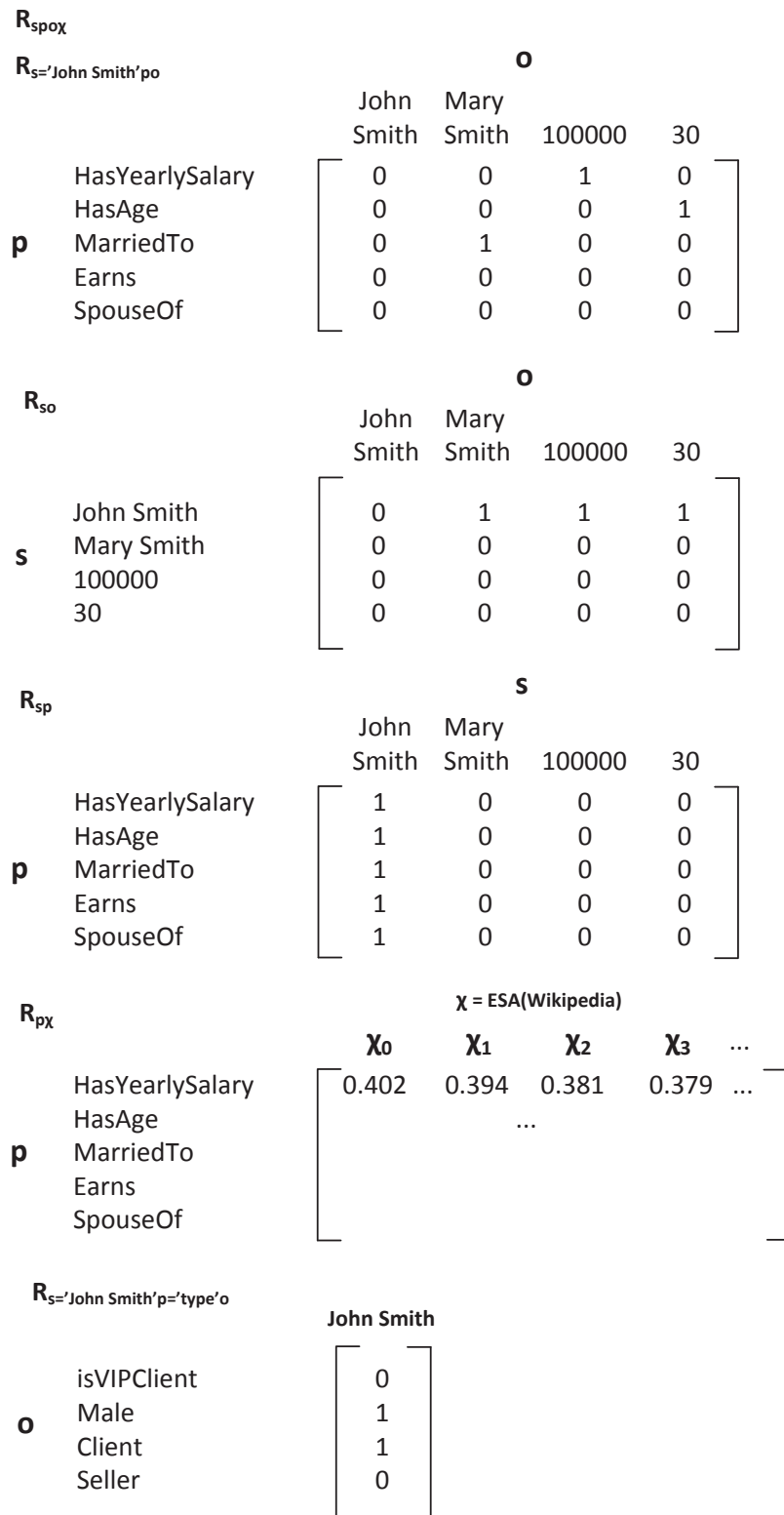
isCustomer(JohnSmith)
hasYearlySalary(JohnSmith, 100000)
hasAge(JohnSmith, 30)
marriedTo(JohnSmith, MarySmith)
hasChild(JohnSmith, AliceSmith)
isClient(JohnSmith)
male(JohnSmith)

Figure 6.16 depicts the dimensions of the distributional tensor for each triple element.

The instantiated tensor for the example knowledge base projected as bi-dimensional matrix components is depicted in Figure 6.17.

6.7 The τ – Space as an Inverted Index

The τ – Space is instantiated as an *inverted index* [197]. The inverted index is a data structure which easily maps to the vector space representation and which supports efficient look-ups of the vectors given the labels of the dimensions.

FIGURE 6.17: Matrix projections of the τ – Space tensor.

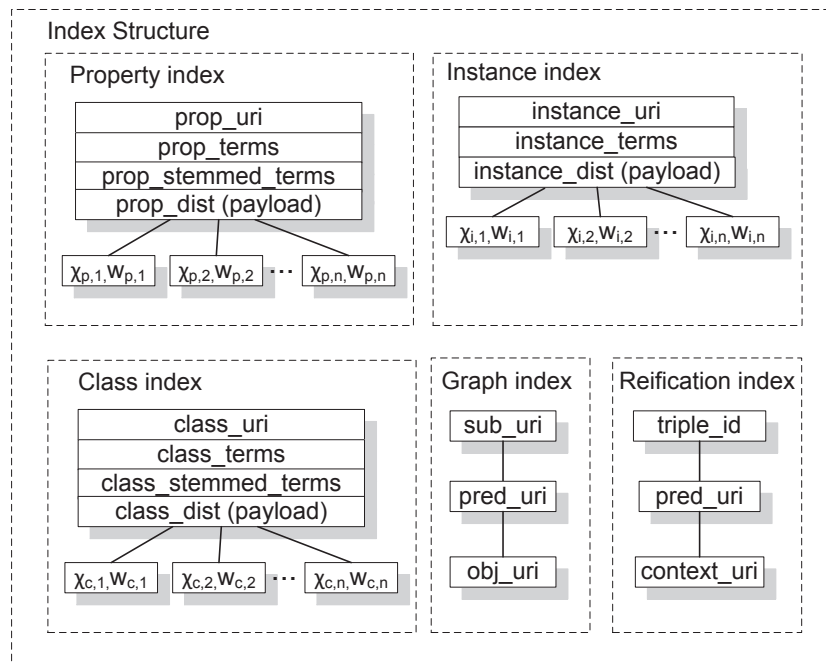


FIGURE 6.18: Distributional inverted index structure.

The core *index* structure consists of five indexes: the *graph index* for mapping the graph topology (triples, and the set of relations), the *instance index* for instances and the *class index* for classes, the *predicate index* and the *reification index*.

While *uri* stores the element URI, the field *terms/stemmed terms* covers the content of the parsed and stemmed URIs (word space) and the *distributional context vector* field, stores the distributional context vectors (distributional space). The distributional context vector is serialized as a *payload* (byte array) under the Lucene framework. The class and the property indexes have associated distributional vectors (Figure 6.18).

The index structure allows its natural distribution and indexing and search parallelization: the separate subspaces can serve as partition identifiers.

6.8 Representation of Complex Classes

6.8.1 Introduction

In this section we describe a lightweight representation approach for complex class descriptors. Complex class descriptors are classes containing more than 3 words for their definition. In order to understand the structure of complex class descriptors, YAGO classes (which map to Wikipedia categories) are analysed. This analysis will support the

construction of a representation model for complex classes. In the representation, complex predicates are decomposed into a graph of terms which can define a compositional-distributional model for complex class descriptors in the context of the τ – Space.

Classes provide names for sets of objects. While some class descriptors are composed of single words or simple expressions (e.g. ‘Person’, ‘Country’, ‘Film’), other descriptors have more complex compositional patterns (e.g. ‘French Senators Of The Second Empire’, ‘United Kingdom Parliamentary Constituencies Represented By A Sitting Prime Minister’).

As the complexity of the domain of discourse, and the decentralization of the content generation, increases in contemporary data management environments, more effort is necessary for defining a consistent and structured conceptual model. As a consequence, as the scale of the domain of discourse increases, data representation strategies move from more structured conceptual models to less structured categorization systems (e.g. *folksonomies*).

This shift from structured towards more unstructured conceptual models is reflected in the content and structure of complex class descriptors. As models get more complex and decentralized, more content is transferred to unstructured natural language descriptors, increasing the terminological variation, reducing the conceptual integration and the structure level of the model. In this scenario, the more formal conceptual model tools are substituted by complex class descriptors as an interface for domain description. From the perspective of information extraction and representation, complex class descriptors provide a much more tractable subset of natural language which can be used as an ‘interface’ for the creation of structured domains. From the syntactic perspective, complex class descriptors are short and syntactically well-formed phrases.

6.8.2 The Structure of Class Descriptors

In order to understand the syntactic structure of complex class descriptors (CCD), an analysis based on the complete set of Wikipedia category links (Figure 6.19) was performed. The complete set contains 287,957 categories. The goal of this analysis is to derive a representation model which can express the relationships between the concepts of the classes following a Semantic Web compatible graph data model. The analysis process started with the manual analysis and categorization of a random sample of 10,000 categories, in order to derive a set of recurrent representation (features) present in the query. Table 6.2 shows the set of category features and instances of categories.

Categories: Barack Obama | 1961 births | Living people | Obama family | 20th-century American writers | 21st-century Am
 Writers from Chicago, Illinois | 20th-century scholars | 21st-century scholars | African-American academics | African-Ameri
 University of Chicago Law School faculty | African-American United States presidential candidates | African-American Unite
 Democratic Party (United States) presidential nominees | Democratic Party Presidents of the United States | Democrati Pa
 Politicians from Chicago, Illinois | Presidents of the United States | United States presidential candidates, 2008 | United Sta
 American people of English descent | American people of Irish descent | American people of Kenyan descent | American p
 Nobel Peace Prize laureates | Recipients of the Presidential Medal of Distinction of Israel | Columbia University alumni | Ha
 International opponents of apartheid in South Africa | Politicians from Honolulu, Hawaii | United Church of Christ members
 American male writers | African-American politicians

FIGURE 6.19: Excerpt of the Wikipedia category links associated with the ‘Barack Obama’ article.

Features	Category Examples
Classes with verbs	United Kingdom Parliamentary Constituencies Represented By A Sitting Prime Minister, Local Government Districts Created By The Local Government Act 1858
Classes with temporal references	19th-century Presidents Of The United States, Tennis Players At The 1996 Summer Olympics
Classes with named entities	Olympic Gold Medalists For The United States, Populated Places In North Holland
Classes with conjunctions	Former Buildings And Structures Of The City Of London, Alumni Of The School Of Oriental And African Studies
Classes with disjunctions	Nobel Laureates In Physiology Or Medicine, Snow Or Ice Weather Phenomena, Converts To Christianity From Atheism Or Agnosticism
Classes with operators	Dutch Top 40 Number-one Singles, World No.1 Tennis Players, Ships Of The First Fleet, Cricketers Who Have Played For More Than One International Team

TABLE 6.2: Core feature set and examples of categories with different feature types.

# of Features	Operators	Words	Proper Nouns	Nouns	Adjectives	Verbs
0	99.846%	0%	46.348%	1.461%	62.284%	81.808%
1	0.154%	15.818%	46.594%	40.173%	32.089%	17.373%
2		26.618%	6.794%	39.727%	5.078%	0.814%
3		24.507%	0.226%	14.572%	0.504%	0.004%
4		18.612%	0.036%	3.339%	0.043%	0.001%
5		8.298%	0.001%	0.610%	0.002%	-
6	-	3.078%	-	0.099%	-	-
≥ 7	-	1.498%	-	0.019%	-	-

TABLE 6.3: Distribution and examples of classes with different feature types.

The manual analysis showed an enumerable set of recurrent features in the class descriptors. After the determination of the core representation features, we automatically analysed the complete set of 287,957 descriptors, according to the incidence of the features. Table 6.3 shows the distribution of features in the full category set. The typical descriptor consists of an entity described by two or more words with one or more specialization relations and it mainly consists of one or more nouns specialized by an adjective. There is, however, a significant variability in the combination of the features set present at the category collection.

FIGURE 6.20: Long tail distribution of POS Tag sequences for Wikipedia categories.

POS Tag Sequence	%
NNS IN NNP	10.05%
NN NNS	7.56%
JJ NNS	7.35%
NN NNS IN NNP	4.68%
JJ NNS IN NNP	4.13%
NNP NNS	3.94%
JJ NN NNS	3.48%
CD NNS	2.63%
NN NN NNS	2.61%
NNS IN JJ NNP	2.00%
NNP NN NNS	1.88%
NN NNP NNS	1.88%
NNS IN JJ	1.80%
JJ NNP NNS	1.71%
NN VBD NNS	1.70%
NNS VBD CD	1.64%
NNS	1.58%
NNP NNS NNS	1.50%
NNP NNS IN NNP	1.43%
NNS IN NN	1.40%
NNS IN NNP NN	1.33%
JJ JJ NNS	1.18%
NNS IN NN NN	1.16%
NNS IN NNP NN NNP	1.09%
Long tail	30.29%

TABLE 6.4: Distribution and examples of classes with different POS Tags.

The possible combination of features follows a long tail distribution which is expressed in the distribution of the sequence of POS Tags for the categories (Figure 6.20). A total of 96 distinct POS Tag sequences were found.

This work concentrates on the use of Wikipedia category links for the analysis of the characteristics of class descriptors. The scale, decentralization and domain variety of Wikipedia categories makes it an ideal resource for the investigation of class descriptors under a high variety scenario. We believe that most of the results from this work can be transported to other complex categorization systems. Similar features and patterns can be observed in other examples of complex CCDs, for example in a domain specific scenario such as the IFRS taxonomy²⁰ and the US GAAP Taxonomy²¹. Examples of categories are: *Franchised Units*; *Partially Owned Properties*; *Residential Portfolio Segment*; *Assets arising from exploration for and evaluation of mineral resources*; *Key management personnel compensation, other long-term employee benefits*.

²⁰<http://www.ifrs.org/>

²¹<http://xbrl.us/taxonomies/>

6.8.3 Representation Model

6.8.3.1 Overview

The representation model is aimed towards facilitating the fine-grained integration between different class descriptors, providing the creation of an integrated and more structured model from the category descriptors. The representation also has an associated interpretation model which aims at making explicit the algorithmic interpretation of the descriptor in the integrated graph.

6.8.3.2 Representation Elements

A class can be segmented into 7 representation elements:

- *Entity*: Entities inside a class descriptor are terms which are terms/entities of the original category which can describe predicates or instances. The entities map to a subset of the content words (open class words), which carry the main content or the meaning of a CCD. Words describing entities can combine *nouns*, *adjectives* and *adverbs*. The entities for an example class descriptors ‘*Snow Or Ice Weather Phenomena*’ are ‘*Snow*’, ‘*Ice*’, ‘*Weather Phenomena*’. Entities are depicted as e_i in Figure 6.21(1).
- *Class & Entity core*: Every entity will contain a semantic nucleus, which corresponds to the phrasal head and which provides its core meaning. For the predicate ‘*Snow Or Ice Weather Phenomena*’, ‘*Phenomena*’ is the class & entity core. Depicted as ‘*’ in Figure 6.21(5).
- *Relations*: Relation terms are binary predicates which connect two entities. In the context of predicate descriptors, relation terms map to closed class words and binary predicates, i.e. prepositions, verbs, comparative expressions (*same as*, *is equal*, *like*, *similar to*, *more than*, *less than*). Depicted as p_i in Figure 6.21(1).
- *Specialization relations*: Specialization relations are defined by the relations between words w_i in the same entity, where w_{i+1} is specialised by w_i . Representing by an unlabeled arrow in Figure 6.21(4).
- *Operators*: Represents an element which provides an additional qualification over entities as a unary predicate. Operators are specified by an enumerated set of terms which maps to adverbs, numbers, superlative (suffixes and modifiers). Quantifiers: e.g. *one*, *two*, *many (much)*, *some*, *all*, *thousands of*, *one of*, *several*, *only*, *most of*; modal: e.g. *could*,

may, shall, need to, have to, must, maybe, always, possibly; superlatives: e.g. largest, smallest, top most; ordinal: 1st, second. Depicted in Figure 6.21(2).

- *Conjunctions & Disjunctions*: A disjunction between two elements ($w_i \vee w_{i+1}e_j$) is defined by the distribution of specialization relations: e_j is specialized by w_i and e_j is specialized by w_{i+1} . A conjunction is treated as an entity which names the conjunction of two entities through a conjunction labeled link. The conjunction representation is depicted in Figure 6.21(2,4).
- *Temporal Nodes*: Consists in the representation of references to temporal elements into a normalized temporal range format.

The representation elements previously described are defined below.

Let *Stopwords* be a set of stopwords that are not used in the representation model. For each complex category cl , we associate the set $Terms(cl)$ formed by all **relevant terms of cl** , that is, $Terms(cl) = \{t : t \in (cl \setminus Stopwords)\}$.

The set $Terms(cl)$ can be split into the following disjoint sets:

- $Ent(cl)$ is formed by nouns, adjectives and adverbs. The terms in $Ent(cl)$ are called **atomic terms** and the elements that provide the core meaning of a complex category are called **entity nucleus** and will be denoted by t^* ;
- $Rel(cl)$ is formed by prepositions, verbs and comparative expressions which represent the relations presented in cl ;
- $Oper(cl)$ is formed by operators;
- $Temp(cl)$ is formed by temporal elements. Temporal elements are normalized into $(dd_i/mm_i/yyyy_i - dd_f/mm_f/yyyy_f)$ representing a time interval starting in $dd_i/mm_i/yyyy_i$ and ending in $dd_f/mm_f/yyyy_f$.

Definition 6.11. A complex category cl is represented by a graph $G(cl)$ defined as an injective total function

$$G(cl) : I \rightarrow 2^{(N \cup I) \times R \times (N \cup I)}$$

where:

- $N = Ent(cl) \cup Oper(cl) \cup Temp(cl)$ is a set of nodes;

- I is a set of identifiers;
- $R = Rel(cl) \cup Rel_{gen}$, is a set of relations where $Rel_{gen} = \{is_specialized_by, op, time\}$.

Note that in definition 6.11, the identifiers in I are used to identify a set of triples, instead of individual triples. In a graph $G(cl)$ we can have the following types of triples:

- Basic triple: (x, r, y) such that $x, y \in Ent(cl)$ and $r \in R$
- Reified triple: (x, r, y) such that $x, y \in Ent(cl) \cup I$, with one of them belonging to I , and $r \in R$
- Temporal (basic or reified) triple: $(x, time, y)$ such that $x \in Ent(cl) \cup I$ and $y \in Temp(cl)$
- Operator (basic or reified) triple: (x, op, y) such that $x \in Ent(cl) \cup I$ and $y \in Oper(cl)$

The interpretation of each triple is based on an infinite set U of Universal Resource Identifiers (URIs). Each element $x \in Terms(cl) \cup I \cup Rel_{gen}$ is interpreted as $[[x]] \in U$. Thus a triple $tr = (x, r, y)$ is interpreted as $[[tr]] \in U^3$.

In a graph $G(cl)$, the **complete path** P is the sequence of sets of identifiers $\langle S_{id_1}, S_{id_2}, \dots, S_{id_n} \rangle$ such that:

- $\forall id \in S_{id_1}$, id identifies a basic triple $tr = (t^*, r, y)$ where t^* is a term nucleus;
- $\forall id \in S_{id_i}$, all identifiers id' that appear in triples $tr \in id$ are such that $id' \in S_{id_{i-1}}$

Example 6.1. Consider the complex category cl_1 :

20th-centuryRulersOfConstituentOrUnrecognizedStatesInNorthAmerica

The relevant terms $Terms(cl_1)$ are formed by:

- $Ent(cl_1) = \{North, America^*, States^*, Constituent, Unrecognized, Rulers^*\}$
- $Rel(cl_1) = \{of, in\}$
- $Temp(cl_1) = \{01/01/1900 - 31/12/2000\}$

Let $I = \{t_1, t_2, t_3, t_4, t_5\}$ be the set of identifiers where:

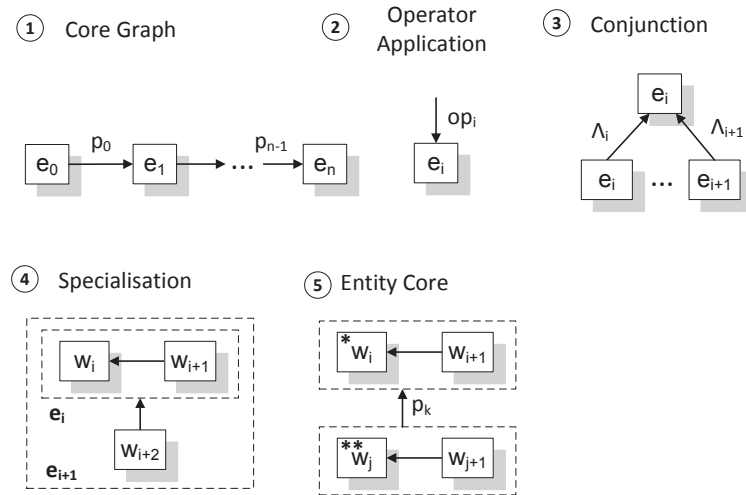


FIGURE 6.21: Graph patterns showing the relations present in the graph representation.

- $t_1 : \text{NorthAmerica}$
- $t_2 : \text{StatesInNorthAmerica}$
- $t_3 : \text{ConstituentOrUnrecognizedStatesInNorthAmerica}$
- $t_4 : \text{20th-centuryRulers}$
- $t_5 : \text{20th-centuryRulersOfConstituentOrUnrecognizedStatesInNorthAmerica}$

The graph $G(cl_1)$ is defined as:

- $t_1 = \{(\text{America}^*, \text{is_specialized_by}, \text{North})\}$
- $t_2 = \{(\text{States}^*, \text{in}, x) \mid x \in t_1\}$
- $t_3 = \{(x, \text{is_specialized_by}, \text{Constituent}), (x, \text{is_specialized_by}, \text{Unrecognized}) \mid x \in t_2\}$
- $t_4 = \{(\text{Rulers}^*, \text{time}, 1900 - 2000)\}$
- $t_5 = \{(x, \text{of}, y) \mid x \in t_4 \text{ and } y \in t_3\}$

and the complete path is $P = \langle \{t_1, t_4\}, \{t_2\}, \{t_3\}, \{t_5\} \rangle$

Depiction of the complex classes from the YAGO dataset are depicted in Figure 6.22.

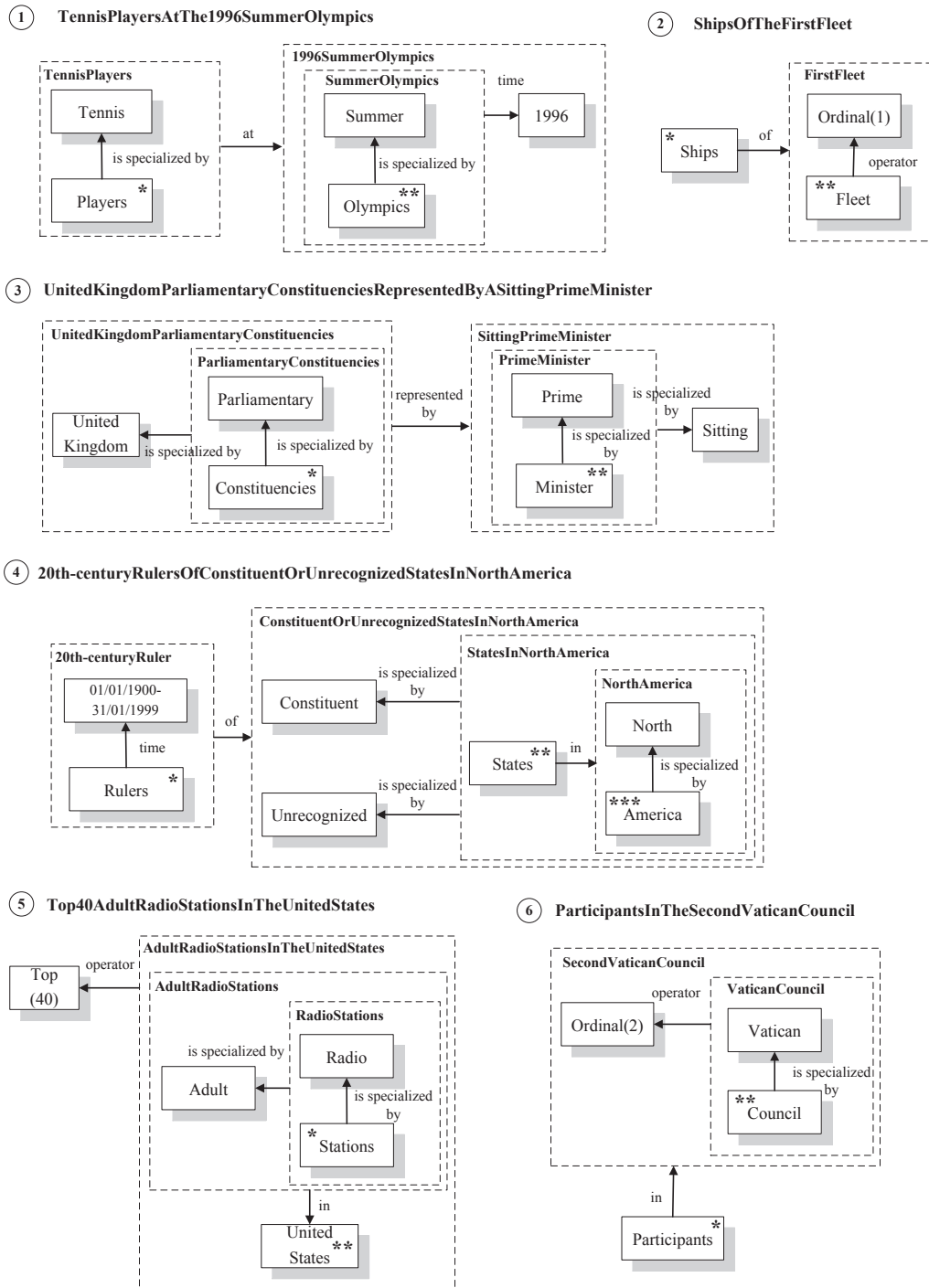


FIGURE 6.22: Categories following the representation model.

6.8.4 Extending the τ – Space for Complex Class Descriptors

The complex class descriptor subspace has a similar structure to the τ – Space (Figure 6.23) In this subspace most of the relations are specialization relations. The dimensional

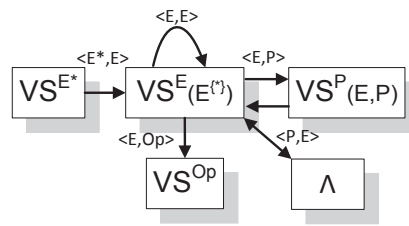


FIGURE 6.23: Depiction of the structure of the complex class subspace.

reduction mechanism is given by the parametrization of the subspaces, following the interpretation from the more generic term, which defines the core entity.

6.9 Chapter Summary

This chapter formalized the definition of the τ – Space, a semantic representation for the *hybrid distributional-relational model*. The τ – Space is a structured vector space model emerging from the embedding of a data graph into a distributional semantics vector space. At the τ – Space, each element in the data graph has an associated distributional semantics vector representation, which supports a geometric-based semantic approximation model, using the distributional knowledge on a large-scale reference corpora. The structure of the τ – Space is defined by the mapping between *data model categories* and the associated distributional vector subspaces associated with each category. Associated publications to this chapter are [177, 198, 199, 200, 201, 202, 203].

Chapter 7

Distributional Semantic Search

7.1 Introduction

The computation of *distributional semantic relatedness measures* over the data elements embedded in the τ – *Space* is at the core of the semantic approximation proposed in this work. The process of using distributional semantic relatedness measures for finding the database entity which is mostly related to a query term can be interpreted as a *distributional semantic search* process over database entities. This chapter focuses on the description of a semantic search approach based on distributional semantic relatedness measures, as semantic search and semantic approximation is at the center of the proposed query approach. This discussion is extended in Chapter 8, where the distributional semantic search is used in coordination with the graph structure and the topology of the τ – *Space* in the form of a *query processing algorithm*.

This chapter is organised as follows: Section 7.2 provides a description of different distributional semantic models that are evaluated in the scope of this thesis; Section 7.3 provides an evaluation of the suitability of different distributional semantic relatedness measures in comparison to WordNet-based semantic similarity and relatedness measures; Section 7.4 defines the distributional semantic search, analyzing how the semantic relatedness measure can be used as a ranking function in the context of a distributional vector space model.

7.2 Distributional Semantic Models

7.2.1 Selecting the Distributional Semantic Models

Different types of distributional semantic models (DSMs) are available in the literature. This section provides a preliminary evaluation of the suitability of different DSMs in the context of this work, using a subset of the core requirements which are impacted by the semantic matching approach (Section 1.6): (i) accurate & comprehensive semantic matching; (ii) low setup & maintainability effort; (iii) interactive search & low query-execution time; (iv) high scalability.

Five models were short-listed from an initial set of high-performing DSMs, based on existing literature evaluation [204], [205], [172], [171] and [53] taking into account their performance for the computation of semantic similarity and relatedness measures.

- Dependency Vectors (Pado & Lapata, 2007) [204].
- Distributional Memory (Baroni & Lenci, 2010) [205].
- Random Indexing (Karlsgren & Salhgren, 2001) [172].
- Latent Semantic Analysis (Landauer & Dumais, 1997) [171].
- Explicit Semantic Analysis (Gabrilovich & Markovitch, 2007) [53].

Dependency Vectors use *syntactic dependencies* as context-filtering functions [204] while Distributional Memory [205] use syntactic dependencies as a context-typing function. Since the dependency on large-scale syntactic parsing of large-scale Web corpora can impact the ability of building transportable distributional models (e.g. for languages or specific subdomains which are not supported by annotated corpora or parsers), impacting the comprehensive semantic matching requirement, and also increasing the setup effort, this work concentrates on DSMs which are not dependent on syntactic or lexicocategorical features.

This section describes the three remaining candidate DSMs. The DSMs are evaluated and compared against WordNet-based semantic relatedness baselines in Section 7.3.2.

7.2.2 Latent Semantic Analysis (LSA) (Landauer & Dumais, 1997)

Latent Semantic Analysis (LSA) [171], is a DSM which focuses on addressing the dimensionality problem by the application of Singular Value Decomposition (SVD) in the

distributional matrix associated with the model. The application of SVD over the distributional matrix, results in a reduction of the dimensionality of the original term vector space, where each dimension after the transformation is represented by a latent dimension. The semantic relatedness measure is computed by the application of the cosine distance between two terms or passages in the LSA space.

LSA has the following configuration parameters:

- \mathcal{C} = documents.
- \mathcal{W} = log term frequency and term entropy in the corpus.
- \mathcal{M} = word \times document.
- d = SVD.
- \mathcal{S} = cosine.

7.2.3 Random Indexing (RI) (Karlsgren & Salhgren 2001)

Random Indexing (RI) [172] is a DSM which uses statistical approximations of the full word co-occurrence data to do dimensionality reduction which results in a performance improvement and fewer required dimensions. The dimensionality reduction is based on Kanervas [206] work on sparse distributed memory.

RI instead represents co-occurrence through *index vectors*. Each word is assigned a high-dimensional, random vector that is known as its index vector. These index vectors are very sparse, which ensures that the chance of any two arbitrary index vectors having an overlapping meaning (i.e. a cosine similarity that is non-zero) is very low.

- \mathcal{C} = context rectangular window-based.
- \mathcal{W} = various.
- \mathcal{M} = matrix word \times word.
- d = RI.
- \mathcal{S} = various.

7.2.4 Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch, 2007)

Explicit Semantic Analysis (ESA) [53] is a distributional semantic model based on Wikipedia. ESA represents the meaning of a text in a high-dimensional space of concepts derived from the Wikipedia text collection. In ESA, the distributional context window is defined by the Wikipedia article, where the context identifier is a Wikipedia article title/identifier.

A *universal ESA space* is created by building a vector space containing Wikipedia articles' document representations using the TF/IDF weighting scheme. In this space, each article is represented as a vector where each component is a weighted term present in the article. Once the space is built, a keyword query over the ESA space returns a list of ranked articles titles, which define a context vector associated with the query terms (where each vector component receives a relevance weight).

In the ESA model, the context is defined at the document level which defines a semantic model which captures both *syntagmatic* and *paradigmatic* relations, appropriate for the computation of a semantic relatedness measures for the schema-agnostic scenario. The coherence of the Wikipedia content discourse in the context of a Wikipedia article also influences the quality of the semantic relatedness measure.

The approach proposed by Gabrilovich & Markovitch also supports a simple compositionality model allows the composition of vectors for multi-word expressions, where the final concept is the centroid of the vectors representing the set of individual terms. The ESA semantic relatedness measure between two terms is calculated by computing the cosine similarity between two distributional vectors.

The link structure of the articles can be used for providing alternative or related expressions for the contexts (based on the extraction of anchor texts) and for the enrichment of the semantic model. The link structure can also work as a basis for dimensional reduction. Gabrilovich & Markovitch describe two levels of semantic interpretation models. *First-order* interpretation models are purely based on information present in the textual description of articles, while *second-order* models also include knowledge present in inter-article links.

Gabrilovich & Markovitch incorporate concept relations by boosting the weights of concepts linked from the top-k weight concepts. The authors apply a further generality filter, where only more general concepts extracted from links are considered. Generality is determined by the difference in the number of inlinks among two linked concepts.

Since some articles are overly specific or are not completely developed, Gabrilovich & Markovitch prune some concepts based on heuristics of quality and relevance.

ESA has the following configuration parameters:

- \mathcal{C} = Wikipedia article.
- \mathcal{W} = TF/IDF.
- d = link-based pruning (optional).
- \mathcal{S} = cosine.

There are notable extensions to the ESA model targeting the generalization of the approach to generic document structures. Polajnar et al. [207] describes two approaches for including document similarity data into ESA, without altering the explicit concept mapping and showing higher correlation with the gold-standard word pair similarities. This work, however, focuses on the ESA model as proposed by [53].

7.3 Evaluating the Suitability of Distributional Semantic Relatedness Measures

7.3.1 Overview

This section evaluates the candidate distributional semantic relatedness measures based on the selected DSMs in comparison with WordNet-based distributional measures. The evaluation aims at verifying the suitability of distributional semantic relatedness measures and their performance relative to WordNet-based measures. This evaluation serves as preliminary evidence for the suitability of distributional semantic relatedness measure as a comprehensive vocabulary mapping and for the selection of the best semantic relatedness measure /distributional semantics model.

7.3.2 WordNet-based measures

Different semantic similarity and relatedness measures based on WordNet were defined in the literature. The list below describes the main semantic similarity and relatedness measures based on WordNet.

Hirst & St-Onge: The work of Hirst & St-Onge [208] proposes a semantic relatedness measure where two concepts x_1 and x_2 are dependent on the length of the path between the synsets of these concepts and of the change in the directionality associated with the relations between entities in WordNet. The relatedness measure is given by:

$$rel_{HS}(x_1, x_2) = C - length(x_1, x_2) - kd$$

where C , k are constants and d represents the number of changes in the direction in the traversal process. $length(x_1, x_2)$ is defined over general (both taxonomic and non-taxonomic) relationship links.

Leacock & Chodrow: Leacock & Chodrow [209] define a measure of similarity dependent on the shortest taxonomic path between the two synsets related to the concepts x_1 and x_2 . The path value is then scaled using the double of the maximum depth D of the taxonomy.

$$sim_{LC}(x_1, x_2) = \frac{-\log(length(x_1, x_2))}{2D}$$

Resnik: This approach [51] is based on the idea that the similarity between two concepts x_1 and x_2 is dependent on the level of the information which is shared by these two concepts. This idea is expressed by the information content of their lowest super-ordinate (L_{Super}). The L_{Super} between two concepts x_1 and x_2 is the most specific concept which is the ancestor of both x_1 and x_2 in the taxonomy. The information content is a measure of specificity associated with the concept and is expressed by the negative logarithm of the probability associated with the concept.

$$sim_R(x_1, x_2) = -\log p(L_{Super}(x_1, x_2))$$

Jiang & Conrath: This approach [210] defines a semantic distance between two concepts x_1 and x_2 based on the idea of information content associated with the concept nodes and with their lowest super-ordinate.

$$dist_{JC}(x_1, x_2) = \log p(x_1) + \log p(x_2) - 2 \log p(L_{Super}(x_1, x_2))$$

Lin: This similarity measure [211] is based on the relation of the information content of the common information between two concepts x_1 and x_2 (defined by the lowest super-ordinate) and the information content associated with the full description of these two concepts.

$$sim_L(x_1, x_2) = \frac{2 \log p(L_{Super}(x_1, x_2))}{\log p(x_1) + \log p(x_2)}$$

Wu & Palmer: Wu & Palmer [212] developed a similarity measure based on the path distance between the concepts and their lowest super-ordinate. In this similarity measure the double of the path length between the lowest super-ordinate and the taxonomy root is divided by the sum of the path lengths between the concepts and their lowest super-ordinate.

$$sim_{WP}(x_1, x_2) = \sum_k w_k \frac{(2 \text{length}(L_{Super}(x_1, x_2), x_0))}{\text{length}(x_1, L_{Super}(x_1, x_2)) + \text{length}(x_2, L_{Super}(x_1, x_2))}$$

where x_0 is the root element of the taxonomy.

Lesk: This approach, proposed by Banerjee & Pedersen [213], measures the semantic relatedness of two terms by computing the number of overlapping words present in the glosses of two WordNet synsets. If the set of overlapping words occur consecutively in the glosses, the squared of the number of words is used.

Vector: This approach, described in [214], forms co-occurrence vectors based on both the glosses and definitions of WordNet concepts. The relatedness of two terms is defined by the cosine of the angle between the co-occurrence vectors.

7.3.3 Comparative Analysis

Previous works in the area of semantic relatedness measures use a correlation with human assessment of the similarity and relatedness of datasets of word pairs as gold-standards to evaluate the performance of the measures. The three main datasets used are:

- *Rubenstein & Goodenough (1965)*[215]: This experiment scored 65 pairs of common nouns in the scale of 0-4 according to their similarity of meaning. 51 subjects participated in the experiment.

- *Miller & Charles (1991)[216]*: In this experiment, 29 of the set of pairs used by Rubenstein & Goodenough were evaluated. The correlation found with the Rubenstein & Goodenough pairs was $\rho = 0.994$. Resnik [51] replicated the same experiment finding a Pearson correlation factor of 0.885, defining a practical upper bound of what semantic relatedness measures can achieve.
- *Finkelstein et al. (2002)[217]*: This experiment, which generated the WordSimilarity-353 dataset, is targeted towards the evaluation of semantic relatedness. The dataset contains two subsets: *set 1* (153 word pairs, evaluated by 13 subjects), and *set 2* (200 word pairs evaluated by 16 subjects) each one containing pairs from different parts-of-speech, a proper noun and pairs involving subjective bias. Selected subjects had near native command of English. The 30 pairs shared with the Miller & Charles experiment have $\rho = 0.939$.

In order to better replicate the type of comparisons used in the context of query-database matching, which include query-dataset alignments which have paradigmatic and syntagmatic relations, this work introduces a new dataset:

- *DBR*: The DBpedia Relatedness dataset was built to provide a complementary perspective of the semantic relatedness grounded on terminological-level data present on the DBpedia ontology. The dataset was built selecting properties and classes from DBpedia and manually associating related natural language terms to each DBpedia term. The types of relations expressed in the construction of the dataset represent relations of different types, mimicking the type of semantic relatedness present in the query-database semantic matching. 118 word pairs were defined. The conditions of the relatedness experiment were similar to the Finkelstein et al. experiment, counting with the same number of participants, with similar profile (graduate students and research staff with native or near native domain of English) and following a similar set of instructions. Each participant was asked to score the semantic relatedness of the word pairs in a range of 0-4, where 0 represents no relatedness and 4 represented totally related words. The calculated mean of the 16 ratings defined the DBR human relatedness gold standard. The DBR dataset and all the experimental data can be found in Appendix B.

Table 7.1 shows the Spearman correlation of the measures with human assessments over different datasets. The framework WordNet::Similarity [218] and was used for the computation of WordNet-based measures, EasyESA[219] for ESA and S-Space [220] for LSA and RI.

Measures	MC	WS-353	DBR
Hirst & St-Onge	0.78	0.37	0.36
Leacock & Chodrow	0.75	0.30	0.21
Resnik	0.75	0.33	0.16
Jian & Conrath	0.71	0.17	0.16
Lin	0.72	0.20	0.16
Wu & Palmer	0.76	0.33	0.19
Lesk	0.81	0.41	0.36
Vector	0.92	0.45	0.51
LSA-TASA	0.71	0.56	0.61
LSA-Wikipedia (2006) $d = 300$	0.78	0.60	0.45
RI-Wikipedia (2006) $d = 1500$	0.51	0.35	0.31
ESA-Wikipedia (2006)	0.63	0.85	0.64

TABLE 7.1: Evaluation of the correlation between semantic similarity and relatedness measures and human correlation using the MC, WS-353 and DBR datasets.

The results show that for the datasets, ESA outperformed the other measures for the computation of semantic relatedness for both WS-353 and DBR. WordNet-based measures performed better for the computation of semantic similarity measures.

This evaluation defines ESA as the best performing distributional semantic model, which will be used in the empirical evaluation of distributional semantics on schema-agnostic query scenarios.

7.4 Distributional Semantic Search

7.4.1 Motivation

This section investigates the suitability of the distributional relatedness measure, as a *terminology-level semantic matching mechanism*. In order to address the semantic matching task, the semantic relatedness measure is used as a *ranking function* for a semantic search mechanism. While the ‘*search by distributional semantic relatedness model*’ is valid for any distributional semantic model, this discussion is instantiated in the context of Explicit Semantic Analysis (ESA) distributional model.

7.4.2 Distributional Semantic Relatedness Measure as a Ranking Function

The semantic relatedness measure s is a real number which quantifies the degree of semantic relatedness between two terms. The computation of semantic relatedness is typically performed pairwise, as the computation of a distance measure in the vector space between two context vectors representing words or terms ($s : W \times W \rightarrow \mathbb{R}$).

The distributional semantic search over a set of indexed terms T and for a query term q can be defined as $\zeta_{V,ST\{dist\}}(q, T, \eta_t)$, where η_t is a semantic relatedness filtering threshold, in which elements above the threshold are considered semantically relevant.

Algorithm 2 describes the naive algorithm for computing the semantic relatedness between the query term q and the set of terms T , by computing the pairwise semantic relatedness.

$\zeta('winston\ churchill', T, 0.004)$		$\zeta('love', T, 0.004)$	
q	T	q	T
winston churchill	war, $\phi = 0.022$ history, $\phi = 0.007$ documentary, $\phi = 0.005$ romantic, $\phi = 0.003$ comedy, $\phi = 0.002$ adventure, $\phi = 0.001$ terror, $\phi = 0.001$ erotic, $\phi = 0.000$	love	romantic, $\phi = 0.020$ erotic, $\phi = 0.004$ comedy, $\phi = 0.004$ documentary, $\phi = 0.003$ war, $\phi = 0.003$ adventure, $\phi = 0.002$ history, $\phi = 0.002$ terror, $\phi = 0.000$

FIGURE 7.1: Distributional semantic search example.

Algorithm 2 Distributional semantic search : $\zeta_{VST\{dist\}}(q, T, \eta_t)$

T : set of terms which are indexed.
 q : query term.
 η_t : threshold.
 $VST\{dist\}$: distributional vector space.
for all $t \in T$ **do**
 $\phi \leftarrow s_{VST\{dist\}}(\vec{q}, \vec{t})$
 if $\phi \geq \eta_t$ **then**
 $R_T \leftarrow (t, \phi)$
 end if
end for
 $rank(R_T)$

where $rank(R_T)$ ranks the result set by the semantic relatedness measure ϕ .

Example:

Figure 7.1 shows the ranked result set for the distributional semantic search for the example queries $\zeta('winston\ churchill', T, 0.04)$ and $\zeta('love', T, 0.04)$ over the predicate/term set

$\{romantic, terror, documentary, erotic, adventure, history, war, comedy\}$. The DSM used is ESA over Wikipedia 2013.

The two queries returns a list of ranked predicates from the most related to most unrelated according to the DSM. A threshold $\eta_t = 0.04$ is applied.

7.4.3 Inverted-index Distributional Semantic Search

While Algorithm 2 provides a description of the functionality behaviour expected by the semantic search, the algorithm can be optimized using an *inverted index data structure*.

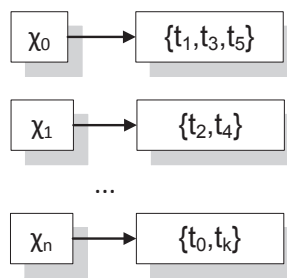


FIGURE 7.2: Inverted index representation for the distributional semantic space.

In information retrieval, an *inverted index* (also named *postings file*) is a data structure which stores the mapping from a content to its location. In document search an inverted index maps from a term to the set of documents in which the term occurs. The inverted index is defined by a tuple that contains the postings for a term into the document collection and the associated weight of the term in relation to the document. Similarity search algorithms consult the postings of each query term to compute similarity scores of documents that have terms in common with the query [221]. Inverted indexes map the structure of the vector space and support the efficient computation of the similarity scores for the vectors.

The inverted index can be used to represent the distributional vector space, where terms representing dimensions are substituted by distributional context vectors, and documents represent terms mapping to database entities (Figure 7.2).

The time cost of a similarity search algorithm is typically dominated by I/O access to the inverted index. Similarity algorithms use a queue to record the current top- k scoring documents. A queue can be implemented with different sorting data structures requiring $O(\log(l))$ comparisons, where l is the length of the queue [221].

Different algorithms can be used to compute vector similarity over an inverted index. The inverted index supports the avoidance of unnecessary query/document similarity comparisons. The similarity search algorithm processes the postings for each query term sequentially, where the scores for each document have a partial tracking. Other similarity algorithms are available, such as *partial ranking* [221]. Maintaining partial scores in a sparse data structure can be used to eliminate some dimensions from the comparison. Partial ranking addresses the space costs of the inverted index search algorithm by eliminating low score documents from the similarity [221]. Techniques for total similarity such as *parallel merge* and *block processing* [221] were also proposed.

Algorithm 3 Distributional semantic search over an inverted index

```

 $Q^x$  : query term distributional vector.
scores : [0, ..., 0]
queue : (term, score) ordered by score (ascending).
for all  $\chi \in Q^x$  do
   $T^x \leftarrow getTerms(\chi)$ 
  for all  $(t, \phi) \in T^x$  do
    scores[t]  $\leftarrow$  scores[t] +  $\phi$ 
    if (length(queue) = k + 1) then
      pop(queue)
      insert((t,scores[t]), queue)
    end if
  end for
end for
pop(queue)
rank(queue, descending)

```

7.4.4 Building the Semantic Space

The procedure for the semantic space construction starts by the construction of the distributional semantics model, which varies for each DSM. Each distributional semantic model has three main core operations which define the interaction with the DSM (Figure 7.3):

1. **computation of the semantic relatedness:** Receives as an input $term_1$ and $term_2$ and returns a real number.
2. **get the context vector for a term:** Receives a $term$ as an input and returns *context vector*.
3. **get related terms:** Receives a $term$ and a *semantic relatedness threshold* and returns a set of ranked terms and associated semantic relatedness scores.

The construction of the ESA context/distributional space starts with the indexing of the Wikipedia articles using TF/IDF in a term vector space model, which defines the *ESA interpreter* (term space). The ESA context/distributional space is built for a set of target terms to be semantically indexed. The target terms are sent as queries to the ESA interpreter which returns a *weighted vector of context vectors*. The weighted context vector encodes a distributional representation of the target word meaning. The context vectors for the target terms are used to build the final ESA context/distributional space. The context associated with each vector component generate new dimensions in the ESA context/distributional space. These steps are depicted in Figure 7.5 and Figure 7.4.

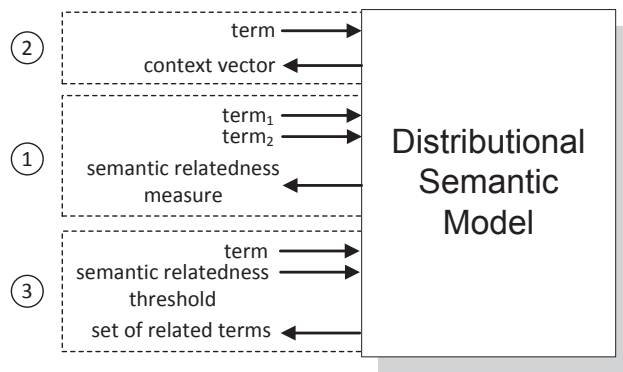


FIGURE 7.3: Interfaces for the interaction with the DSMs.

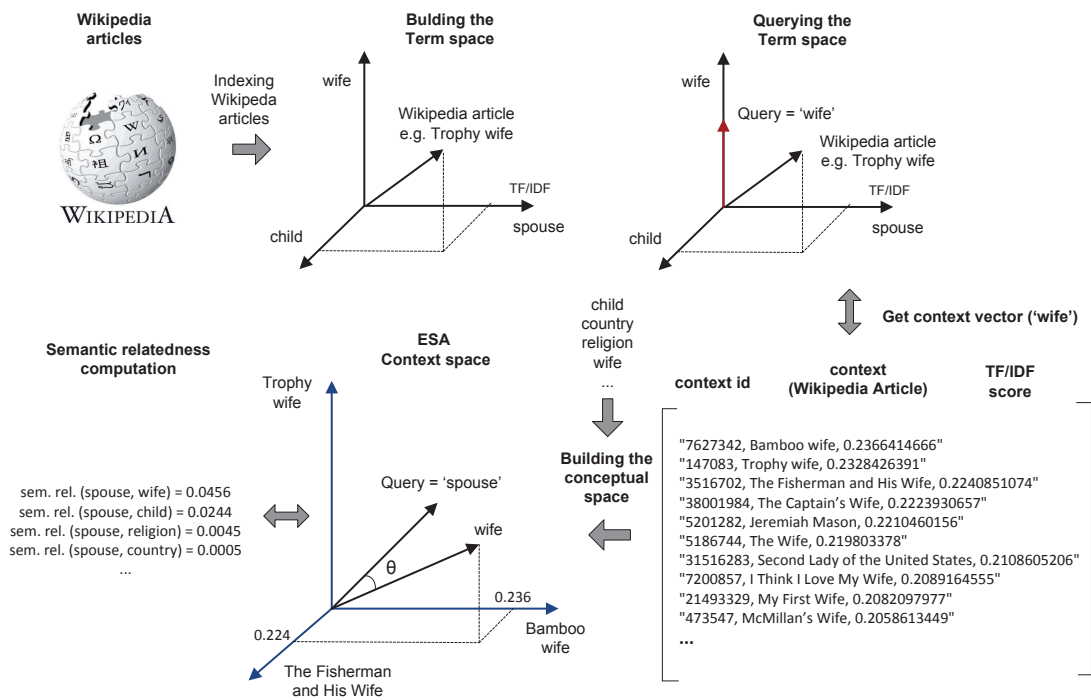


FIGURE 7.4: ESA distributional semantic space construction.

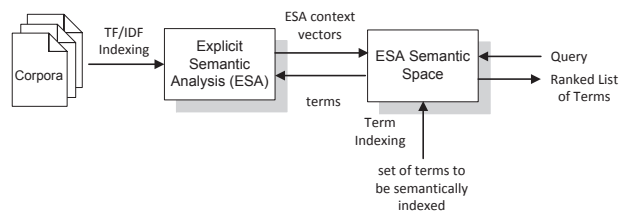


FIGURE 7.5: Workflow for the construction of the ESA distributional space.

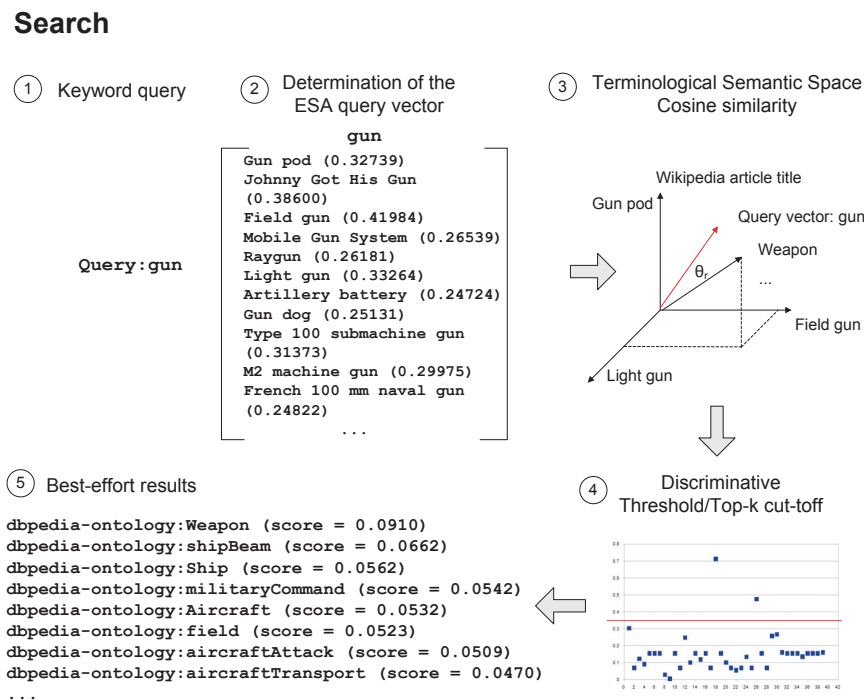


FIGURE 7.6: Terminological semantic space search process.

7.4.5 Searching the Semantic Space

The ESA semantic space forms a vector space which has its dimensionality dependent on the number of indexed terms and on the decision on the dimensionality of the ESA context vectors. In the worst-case scenario, the dimensionality of the terminological space equals the number of Wikipedia articles which are indexed in the ESA semantic vector space.

Figure 7.7 shows reduced ESA context vectors for two example terms: *United States Senators from Illinois* and *spouse*.

Once the distributional semantic search space is built, the semantic search operation can be performed. Figure 7.6 depicts the search process, where for an example keyword query *gun*, the approach returns a list of related concepts (5) from DBpedia. In this case, the target vocabulary concept is the top-most result (*Weapon*). The approach also returns additional terms with some degree of semantic relatedness to *gun*.

The weights associated with the ESA distributional vectors follow a long tail distribution. Figure 7.8 shows the distribution of the weight scores for the context vectors for different words: ‘*power*’, ‘*revenue*’, ‘*love*’ and ‘*tubulin*’. Words with higher specificity have higher scores. The weight distribution can be used to define the cutting point for the size of the distributional weights, based on the specificity-level of a word.

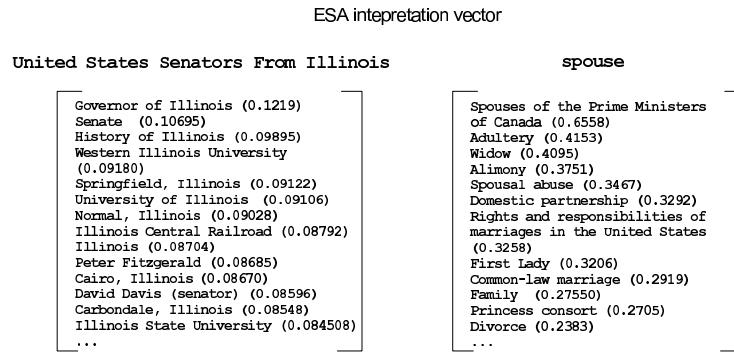


FIGURE 7.7: Examples of ESA interpretation vectors for *United States Senators from Illinois* and *spouse*.

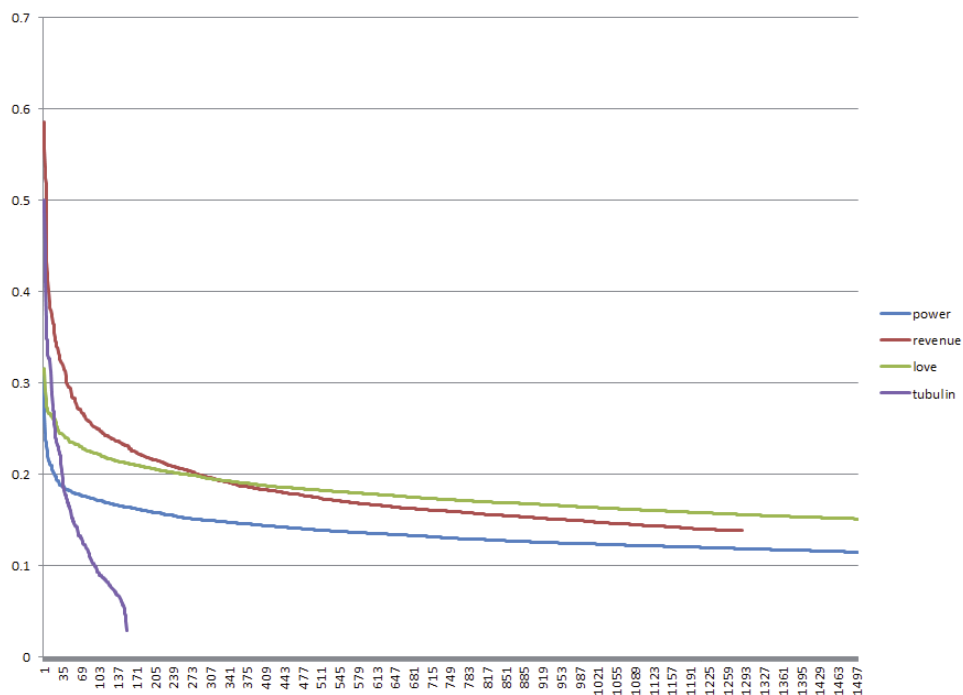


FIGURE 7.8: Values for the weights of the context vectors.

7.5 Evaluating the Distributional Semantic Search: Searching for Database Predicates

7.5.1 Motivation

This section aims at providing a first level evaluation of the distributional semantic search using Explicit Semantic Analysis (ESA) for matching single query terms to terminology-level database terms. In addition to a preliminary verification of the suitability of ESA for terminology-level semantic matching, the section uses the search experiment in the

determination of a *semantic differential model*, to determine a *cut-off threshold* based on the behaviour of the semantic relatedness measure as a ranking function.

7.5.2 Evaluating the Terminology-level Search

The approach was evaluated building a search space indexing 1,610 concepts (275 classes and 1,335 properties) present in the 3.6 version of the DBpedia ontology. The DBpedia ontology was chosen due to the size and comprehensive nature of the ontology and due to its open domain nature. A prototype of the semantic search space was implemented. The prototype was built focusing on measuring the quality of the proposed approach, consisting of an in memory inverted terminological index and an ESA concept space [53]. The 2006 version of Wikipedia (approximately 1.5 million articles) was used in the creation of semantic space and a size of 50 concepts was defined for each concept vector.

The procedure for generating the set of keyword queries was based on the process of asking two users to tag 60 commonsense images and their constituent elements with keywords. The set of tags which could be mapped to related concepts in the DBpedia ontology were used to define the set of 143 keyword queries (query size of 1-2 terms). The queries are available in Appendix C. This procedure was used to generate the *search for highly related concepts* behaviour expected in terminological search, where users are abstracted away from the terms in the ontology. The information present in properties' domains and ranges axioms were not used in the indexing process: just the specific ontology element name embedded in each URI was used. The data output associated with the experiments can be found in Appendix C.

7.5.3 Qualitative Analysis

In order to make the discussion on the semantic matching properties of the terminological space more concrete, examples of keyword queries and best-effort results are listed in Figure 7.9 and Figure 7.10. The example queries lists the top-8 most semantic related terms to natural language queries over the DBpedia ontology. The example queries illustrate the semantic matching problem for terminological search, where the closest related concept can be expressed by different semantic relationships, varying from *string variations* (e.g. books - Book), *synonyms* and *taxonomic ancestors* to *broader classes of semantic relations* (e.g. justice - SupremeCourtOfTheUnitedStatesCase, Judge). The fine grained semantic nature of the search approach is exemplified in the queries *bass* and *bassist*, where the closest related concept *Instrument* is highly ranked in the *bass* query. For the query *bassist* the closest related concept *musician* is highly ranked. These examples demonstrate the reasoning-like behaviour of the semantic approximation

```

Query: [airplane]
dbpedia-ontology:Aircraft (score = 0.1146)
dbpedia-ontology:aircraftAttack (score = 0.1097)
dbpedia-ontology:flyingHours (score = 0.1008)
dbpedia-ontology:aircraftTransport (score = 0.0876)
dbpedia-ontology:aircraftPatrol (score = 0.0738)
dbpedia-ontology:aircraftHelicopterTransport (score = 0.0709)
dbpedia-ontology:aircraftHelicopter (score = 0.0706)

Query: [gun]
dbpedia-ontology:Weapon (score = 0.0910)
dbpedia-ontology:shipBeam (score = 0.0662)
dbpedia-ontology:Ship (score = 0.0562)
dbpedia-ontology:militaryCommand (score = 0.0542)
dbpedia-ontology:Aircraft (score = 0.0532)
dbpedia-ontology:field (score = 0.0523)
dbpedia-ontology:aircraftAttack (score = 0.0509)
dbpedia-ontology:aircraftTransport (score = 0.0470)

Query: [bass]
dbpedia-ontology:musicalBand (score = 0.0962)
dbpedia-ontology:discoverer (score = 0.0721)
dbpedia-ontology:discovered (score = 0.0721)
dbpedia-ontology:Instrument (score = 0.0621)
dbpedia-ontology:instrument (score = 0.0621)
dbpedia-ontology:Band (score = 0.0597)
dbpedia-ontology:associatedBand (score = 0.0597)
dbpedia-ontology:band (score = 0.0597)

Query: [bassist]
dbpedia-ontology:musicians (score = 0.1743)
dbpedia-ontology:lounge (score = 0.0886)
dbpedia-ontology:maidenFlight (score = 0.0685)
dbpedia-ontology:billed (score = 0.0684)
dbpedia-ontology:winsAtAus (score = 0.0632)
dbpedia-ontology:maidenFlightRocket (score = 0.0569)
dbpedia-ontology:musicComposer (score = 0.0509)
dbpedia-ontology:Instrument (score = 0.0484)

Query: [wife]
dbpedia-ontology:monarch (score = 0.0804)
dbpedia-ontology:Monarch (score = 0.0804)
dbpedia-ontology:spouse (score = 0.0764)
dbpedia-ontology:timeZone (score = 0.0707)
dbpedia-ontology:person (score = 0.0610)
dbpedia-ontology:Person (score = 0.0610)
dbpedia-ontology:personName (score = 0.0610)
dbpedia-ontology:foundingPerson (score = 0.0602)

```

FIGURE 7.9: Set of example queries over the DBpedia vocabulary and top-8 results.

behavior which is supported by distributional semantic search. Some of the queries allow the verification of the semantic conjunction behavior where multiple keywords should match the closest related concept for the conjunction of keyword concepts, instead of returning disjoint matches for each keyword query. Figure 7.11 exemplifies this behavior using the queries *engine* and *car engine* and the list of associated rankings.

Additionally, one characteristic which is not fully expressed in the comparative evaluation measures is the fact that the proposed approach provides a more comprehensive exploratory search behaviour, allowing users to have a better understanding of the conceptual coverage of the elements on the vocabularies.

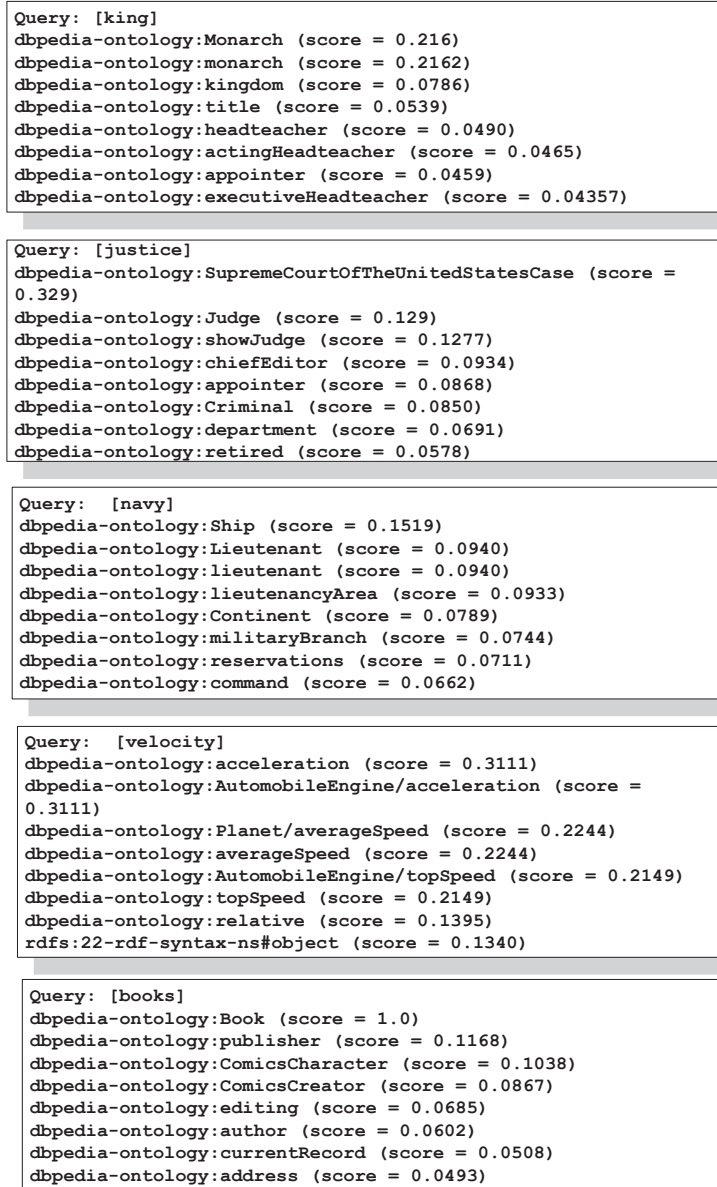


FIGURE 7.10: Additional set of example queries over the DBpedia vocabulary and top-8 results.

```

Query: [engine]
dbpedia-ontology:engine (score = 1.0)
dbpedia-ontology:engine (score = 1.0)
dbpedia-ontology:engineType (score = 0.7174)
dbpedia-ontology:gameEngine (score = 0.6452)
dbpedia-ontology:principalEngineer (score = 0.5306)
dbpedia-ontology:cylinderCount (score = 0.1784)
dbpedia-ontology:cylinderBore (score = 0.17584)
dbpedia-ontology:AutomobileEngine/cylinderBore (score = 0.17584)
dbpedia-ontology:pistonStroke (score = 0.12070)
dbpedia-ontology:AutomobileEngine/pistonStroke (score = 0.12070)
dbpedia-ontology:AutomobileEngine (score = 0.10195)

Query: [car engine]
dbpedia-ontology:carNumber (score = 0.38506)
dbpedia-ontology:AutomobileEngine (score = 0.15942)
dbpedia-ontology:layout (score = 0.12297)
dbpedia-ontology:cylinderBore (score = 0.10015)
dbpedia-ontology:AutomobileEngine/cylinderBore (score = 0.10015)
dbpedia-ontology:cylinderCount (score = 0.0965)
dbpedia-ontology:engineer (score = 0.0944)
dbpedia-ontology:engine (score = 0.0944)
dbpedia-ontology:displacement (score = 0.0921)
dbpedia-ontology:AutomobileEngine/displacement (score = 0.0921)
dbpedia-ontology:secondDriver (score = 0.0876)

```

FIGURE 7.11: Example of the conjunction of two predicates.

7.5.4 Quantitative Analysis

The quantitative part of the evaluation measures the quality of the approach under the scope of the motivations and requirements for terminological search. The first measure, *% of queries correctly answered with semantically related terms*, evaluates the percentage of queries which are answered with resources which are closely semantically related. The results show that the **distributional semantics search approach answers 92.25%** of the 143 queries with semantically related terms. *Average precision@k* is defined as the number of closely related terms in the top-k semantically related results over the number of returned results. The approach presents high average precision, which is kept along the top-5 and top-10 results (**avg. p@5=0.732, avg. p@10=0.691**). *Mean reciprocal rank* (MRR) measures the ranking quality by calculating the inverse of the rank of the best result (for an extensive discussion see Chapter 9). In the case of the list of semantically related results the best-result is defined as the closest semantically related concept and not as the top related ranked result. The final MRR value shows that the quality of the ranking is high (**MRR=0.646**) where, in average, most of best results are located on the first or second positions. Table 7.3 summarizes the results.

In order to provide a comparative baseline for the approach, the same vocabulary was indexed using a TF/IDF index which, under the minimum description assumption of the experiments, worked essentially as a simple stemming-based search. A second baseline

# of queries answered	avg. p@5	avg. p@10	mrr
92.25	0.732	0.691	0.646

TABLE 7.2: Evaluation metrics for the ESA-based terminology-level semantic search.

Approach	# of queries answered
ESA	92.25%
String matching	45.77%
WordNet QE	52.48%

TABLE 7.3: Comparative analysis of the number of queries answered in relation to two baselines: (i) string matching (stemming) and (ii) WordNet query expansion.

was generated using a WordNet-based query expansion. The results show that the *distributional approach largely outperforms the string search and the WordNet-based query expansion approach*, where the **first (simple term search) baseline answers 45.77% of the queries** and the **second (term search + WordNet query expansion) baseline answers 52.48% of the queries**, compared to **92.25% for the distributional ESA approach**.

7.5.5 Analysis of the Distributional Space Dimensionality

Figure 7.12 depicts the growth of the dimensionality of the distributional vector space. For 1,610 predicates the final dimensionality of the space is around 30,000. The growth of the dimensionality of the space is linear with the number of indexed resources. With the increase of the number of properties the growth rate tends to reduce due to the overlap of dimensions between different distributional vectors. For this example, the dimensionality of the space corresponds to 37.2 % of the number of dimensions for all the vectors (overlap of 62.8 %).

7.6 Semantic Differential Analysis

Semantic models capture the semantic relationships between different concepts. In DSMs all semantic associations are 1-level relationships and the degree of semantic association is defined as a distance measure between the vector representations. The understanding of the relationship between the value ranges associated with the distance measures in a specific distributional semantic model and the semantic proximity between two concepts is fundamental to the use of DSMs for semantic approximation tasks. In the context of semantic search, the semantic vector distance, i.e. the semantic relatedness score, is

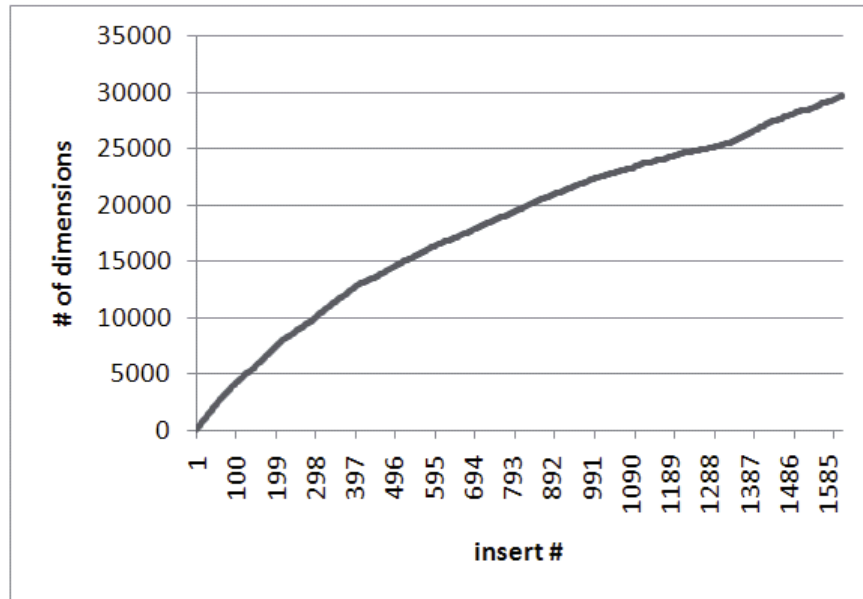


FIGURE 7.12: Growth in the dimensionality of the space by the distributional semantic model.

FIGURE 7.13: Possible nominal classification systems for the semantic relatedness values.

used as a ranking function, which represents the semantic proximity between query and database elements.

For different semantic models, the semantic relatedness value ranges can be mapped into different *nominal values* which define the degree of semantic relatedness between two terms. Figure 7.13 shows three possible classification systems for semantic relatedness values. The definition of nominal values for semantic relatedness facilitates the interpretation and use of the value. Figure 7.13 (2) is the classification system adopted in this work. The score 1.0 represents an identical matching, followed by three categories: *strongly semantic related*, *semantically related*, *semantically unrelated*. The *semantic threshold filter* defines the the cut-off between semantic related and semantic unrelated. A semantic threshold filter can be also applied to the transition between different semantic relatedness categories.

In this section a methodology is described for the definition of the semantic relatedness value ranges. Each combination of *DSM*, *reference corpora* and *application scenario* defines the classification system and the specific range values.

The determination of the *filtering threshold* can be performed by analyzing the behaviour of the derivative of the function defined by the ranked list of semantic relatedness value (Figure 7.14). This analysis, called in the context of this work *semantic differential*

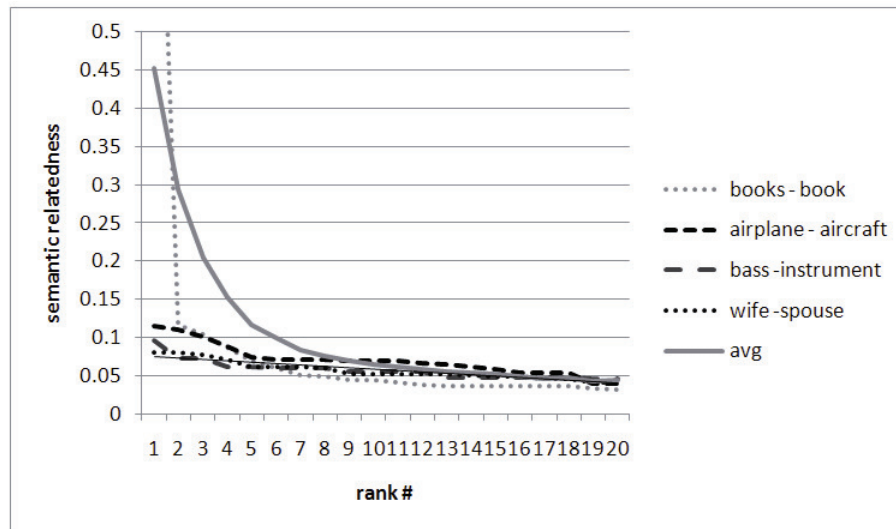


FIGURE 7.14: Semantic relatedness scores for sample query-vocabulary matches.

analysis, aims at understanding the behavior of the semantic relatedness vs. ranking position function to define the mapping to the nominal classification.

Two methods are proposed in this work:

- *Unsupervised*: Consists of the automatic detection of a discontinuity in the ranked list of semantic relatedness value. A semantic gap is a higher deviation in the derivative value between two consecutive ranked items. This method can be *dynamic*, where the threshold value is detected independently for each ranked list, or *static*, where the gap is defined over a finite set of ranked lists.
- *Supervised*: Consists in the manual definition of a training set for the categorization of a set of ranked list of results, where terms in the list are classified according to the set of nominal categories. The training set is used to define an average model for the set of fixed thresholds which maximize the discrimination of the nominal categories.

The *semantic differential analysis* is described below and it has the goal of defining an unsupervised method.

This analysis can support the detection of a semantic gap between highly semantically related resources and the top average non-related terms. The distribution of the top-20 (non-filtered) semantic relatedness scores for 4 queries + the average of the scores for all 143 queries is depicted on Figure 7.14.

Figure 7.15 shows the symbols that are used to describe the main concepts of the *semantic differential model*, depicting a ranked list of results, where S_k represents the

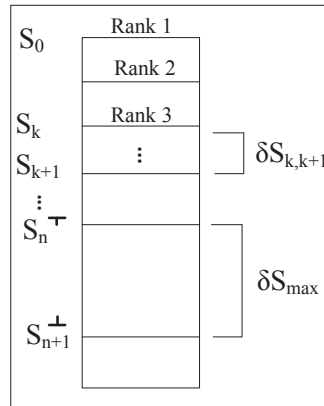


FIGURE 7.15: Depiction of the elements of the semantic differential model over a ranked list of results.

Measure	Value
Avg. Semdiff (δS)	0.006523
Avg. Maximum Semdiff (δS_{max})	0.281752
Avg. Maximum Relatedness Value (S_{max})	0.452145
Avg. Relatedness Value: Top Semdiff Extreme (S_n^\top)	0.417370
Avg. Relatedness Value: Bottom Semdiff Extreme (S_{n+1}^\perp)	0.135618
% of Top Semdiff Extreme (S_n^\top) ≥ 0.1	81%
$0.09 \leq$ % of Top Semdiff Extreme (S_n^\top) < 0.1	4%
$0.07 \leq$ % of Top Semdiff Extreme (S_n^\top) < 0.09	8%
% of Top Semdiff Extreme (S_n^\top) < 0.07	7%
% of Bottom Semdiff Extreme (S_{n+1}^\perp) ≥ 0.1	44%
$0.09 \leq$ % of Bottom Semdiff Extreme (S_{n+1}^\perp) < 0.1	9%
$0.07 \leq$ % of Bottom Semdiff Extreme (S_{n+1}^\perp) < 0.09	18%
% of Bottom Semdiff Extreme (S_{n+1}^\perp) < 0.07	29%

TABLE 7.4: Measures and distribution for the elements semantic differential analysis.

relatedness values associated with the $k+1$ ranked concept, S_0 is the maximum relatedness value, $\delta S_{k,k+1}$ the semantic differential between two adjacent ranked concepts, δS_{max} is the maximum semantic differential in the unfiltered ranked list and S_n^\top , S_{n+1}^\perp are respectively the top and bottom relatedness values of δS_{max} .

Table 7.4 shows the values and the distribution of the elements of the semantic differential model for the full (unfiltered) query/result set. Queries with literal string matching approach semantic relatedness scores close to 1 (the maximum value). On the average, high conceptually related matching happens on the range between 0.5 and 0.1. The average size of the maximum semantic differential is significantly larger than the average semantic differential, showing a clear discriminative nature for the relatedness score. Most of δS_{max} values are located above 0.1. This is confirmed by the distribution of S_n^\top , S_{n+1}^\perp which also shows that very few δS_{max} fall below 0.07. The range 0.1 to 0.07 still represents a significant range for semantically related concepts.

The final threshold $t(S)$ is defined as:

$$t(S) = \begin{cases} S_{n+1}^{\perp} & \text{if } S_n^{\top} > 0.1 \text{ and } S_{n+1}^{\perp} > 0.07 \\ 0.07 & \text{if } S_{n+1}^{\perp} < 0.07 \end{cases}$$

The semantic differential analysis defines a threshold criteria for the relatedness scores. The specific values which define the threshold are specific to ESA and to the corpora used and it is likely that these values will differ for other corpora and distributional models. The main contribution of the differential analysis proposed here is the definition of a principled differential semantic model and threshold determination methodology which can be reused in different distributional models.

7.7 Chapter Summary

This chapter described the distributional semantic search approach in which the *distributional semantic relatedness measure* is used as a *ranking function*. The *semantic differential* approach for the determination of the semantic relatedness-based ranking threshold is introduced, supporting the filtering of unrelated results. The semantic search is evaluated for an open domain terminology-level search scenario, achieving a substantial query coverage improvement when compared to WordNet-based query expansion and simple string-based matching (**simple term search baseline answers 45.77% of the queries** and the **term search + WordNet query expansion baseline answers 52.48% of the queries**, compared to **92.25% for the distributional ESA approach**). Associated publications to this chapter are [222, 223].

Chapter 8

The Schema-agnostic Query Processing Approach

“... computer science is rife with phenomena whose understanding requires close attention to the interaction between language and structure.”

Scott Weinstein, Finite Model Theory
and Its Applications

8.1 Introduction

The semantic model expressed in the τ -Space which was introduced in Chapter 6 serves as the basis for the construction of the schema-agnostic query approach. While the τ -Space corresponds to the distributional knowledge representation model, the *query processing approach* provides the compositional-distributional component which uses the τ -Space model to match query with the elements in the database conceptual model and the set of database operations, working as a query-database semantic matching algorithm.

The query processing approach has the objective of providing a mapping $m(Q, DB)$ between the query terms $\langle q_0 \dots q_n \rangle \forall q_i \in Q$ and E elements in the database DB . This mapping defines an *interpretation* of the query Q under a database DB and the distributional model DSM . The interpretation process aims at minimizing the impact of *ambiguity*, *vagueness* and *synonymy* between Q and DB .

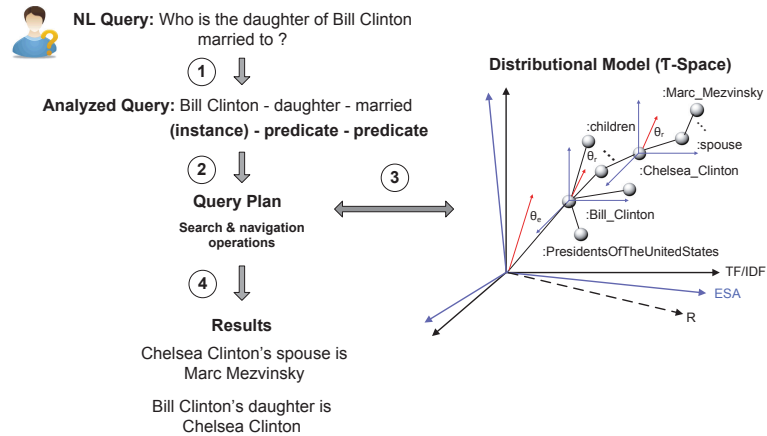


FIGURE 8.1: High-level workflow of the steps of the query processing approach.

This chapter describes the schema-agnostic query processing approach and it is organised as follows: Section 8.2 provides an overview and high-level perspective of the components involved in the schema-agnostic query approach; Section 8.4 describes the SPARQL semantics which serves as a functional reference for the schema-agnostic query; Section 8.5 describes the query analysis component of the approach; Section 8.6 describes the schema-agnostic query processing approach and algorithm, which provides a detailed account of the query-processing model. Finally, Section 8.7 describes the implementation of the schema-agnostic *Treo* system which is used to evaluate the approach.

8.2 Overview of the Schema-Agnostic Query Approach

The query processing approach defines a set of *semantic search*, *entity composition* and *solution modifier operations* over the τ -Space. The τ -Space is built for a given dataset during indexing time. In this section the core query processing workflow is described. The combination of the τ -Space representation with the *query processing* defines the schema-agnostic query model.

Figure 8.1 shows the high level elements and components of the query processing workflow. The query processing workflow starts with the analysis of the natural language query, from which a set of *query features* and a *semi-structured query representation* is extracted (step 1). After the query is analyzed, a *query processing plan* is generated, which maps the set of query features and the semi-structured query into a set of *search*, *entity composition* and *solution modifiers operations* (step 2) over the data graph embedded into the τ -Space. These operations define the semantic matching between the query and the data, using the distributional semantic vector representation. The processed query returns the set of results from the τ -Space (step 4).

8.3 Principles of the Schema-agnostic Query Approach

In order to build the schema-agnostic query mechanism, four main *guiding principles* are employed:

1. **Approximate query model:** The proposed approach targets semantically approximate solutions. Instead of expecting the query mechanism to return exact results as in structured database queries, this work focuses on the highly semantically related results, which can be later filtered by the data consumer. However, an explicit requirement in the construction of an approximate approach for queries over structured data is the *conciseness* of the answer set, where more selective results are targeted, instead of an exhaustive ranked list of results (as in search engines).
2. **Use of distributional semantic relatedness measures to match query terms to dataset terms:** Distributional semantic relatedness and similarity measures allow the computation of a measure of semantic proximity between two natural language terms. While semantic similarity measures are constrained to the detection of a reduced class of semantic relations, and are mostly restricted to compute the similarity between terms which are nouns, semantic relatedness measures are generalized to other types of semantic relations and terms from different lexical categories. This makes them more robust to the heterogeneity dimensions of the vocabulary gap. The use of comprehensive knowledge sources allows the creation of a high coverage distributional semantic model.
3. **Context-based semantic matching:** Consists of the prioritization of the matching of query terms which are less bound to the *ambiguity*, *vagueness* and *synonymy* conditions, using the first alignments as contextual constraints (semantic pivots) and reducing the dimensionality of the matching configuration space and the uncertainty and performance problems associated with it.
4. **Query-dataset compositional correspondence:** The compositional model is given by two types of correspondence: (i) *Part-of-Speech - Entity Type (Data Model category) Correspondence* (ii) *Query Syntactic structure (Phrase structure/Dependencies between phrasal heads) - Subject-Predicate-Object-Context (S-P-O-C) Correspondence*.

8.4 SPARQL Semantics

8.4.1 Motivation

SPARQL is a structured query language for RDF(S). Users querying a structured dataset with schema-agnostic queries expect to find a similar set of operations to the ones provided by the structured query language associated with the data model. This section introduces the SPARQL query language in order to provide a basis of discussion for the operations over the τ – *Space*. In this section the basic elements of a query over an RDF(S) graph are defined. The definitions are based on the SPARQL specification [224]. To define the graph navigation, compositional patterns and the solution modifiers, the SPARQL specification notation is followed [224], SPARQL Algebra [225], Perez [226] and Hartig’s formalisation [227] is used in the definition of a SPARQL iterator based query mechanism over the Linked Data Web.

8.4.2 Basic Definitions

Given a data source DB , a query consists of a graph pattern which is matched against the DB , and the values obtained from this matching are processed to give the answer. The data source DB to be queried can be composed of multiple sources. In SPARQL, a query consists of three parts:

- **Graph Pattern:** Contains the structural part of the query. In SPARQL this maps to the graph patterns and the composition of OPTIONAL, UNION and FILTER operators.
- **Solution Modifiers:** Allows the modification of the result set by applying operators like projection, distinct, order, limit, and offset.
- **Query Form:** Maps to query types, which define different outputs: YES/NO, SELECT, ENTITY DESCRIPTION.

The correspondence between natural language and SPARQL *query forms* is given below:

- **SELECT:** Returns all, or a subset of, the variables bound in a query pattern match.
- **YES/NO:** Returns a boolean indicating whether a query pattern matches or not. Maps to ASK queries in SPARQL.

- **ENTITY DESCRIPTION:** Returns an RDF(S) graph that describes the resources found. Maps to DESCRIBE in SPARQL.

In the scope of this work, we will not investigate the corresponding schema-agnostic mechanism for SPARQL **CONSTRUCT** query form (which returns an RDF(S) graph constructed by substituting variables in a set of triple templates).

The definitions below, based on [224], [225], [226] and [227], define the core constructs of a SPARQL query:

Definition 8.1 (Definition Abstract Query). An *abstract query* is a tuple (E, DS, R) where: E is a query algebra expression, DS is an RDF(S) Dataset and R is a query form.

Definition 8.2 (RDF Term). Let I be the set of all IRIs. Let RDF^L be the set of all *RDF Literals*. Let RDF^B be the set of all blank nodes in RDF graphs. The set of RDF Terms, $RDF - T$, is $I \cup RDF^L \cup RDF^B$.

Definition 8.3 (RDF Dataset). An *RDF dataset* is a set: $G, \langle u_1 \rangle, G_1, \langle u_2 \rangle, G_2, \dots, \langle u_n \rangle, G_n$ where G and each G_i are graphs, and each $\langle u_i \rangle$ is an IRI. Each $\langle u_i \rangle$ is distinct. G is called the default graph. $\langle u_i \rangle, G_i$ are called named graphs.

Definition 8.4 (Active Graph). The *active graph* is the graph from the dataset used for basic graph pattern matching.

Definition 8.5 (Query Variable). A *query variable* is a member of the set V , where V is infinite and disjoint from $RDF - T$.

Definition 8.6 (Triple Pattern). A *triple pattern* is member of the set: $(RDF^T \cup V) \times (I \cup V) \times (RDF^T \cup V)$

Definition 8.7 (Basic Graph Pattern). A *basic graph pattern* is a set of *triple patterns*.

Definition 8.8 (Multiset). When matching graph patterns, the possible solutions form a *multiset*. A *multiset* is an unordered collection of elements in which each element may appear more than once. It is described by a set of elements and a cardinality function giving the number of occurrences of each element from the set in the multiset.

Definition 8.9 (Solution Mapping). A *solution mapping* is a mapping from a set of variables to a set of RDF terms which is a partial function $\mu : V \rightarrow T$. For a triple pattern t , $\mu(t)$ is the triple obtained by replacing the variables in t according to μ . The domain of μ , $dom(\mu)$, is the subset of V where μ is defined.

Definition 8.10 (Compatible Mappings). Two mappings μ_1 and μ_2 are *compatible* when for all $x \in dom(\mu_1) \cap dom(\mu_2)$, it is the case that $\mu_1(x) = \mu_2(x)$.

Definition 8.11 (Filter). Let Ω be a multiset of solution mappings and $expr$ be an expression. Given a mapping μ and a built-in condition R , then μ satisfies R , denoted by $\mu \models R$, if:

$$\Omega \text{ FILTER } R = \{\mu \mid \mu \in \Omega \wedge expr(\mu) = true\}$$

Definition 8.12 (Join). Let Ω_1 and Ω_2 be multisets of solution mappings. We define:

$$\Omega_1 \bowtie \Omega_2 = \mu_1 \cup \mu_2 \mid \mu_1 \in \Omega_1 \wedge \mu_2 \in \Omega_2 \wedge \mu_1 \wedge \mu_2 \text{ are compatible}$$

Definition 8.13 (Difference). Let Ω_1 and Ω_2 be multisets of solution mappings. We define:

$$\Omega_1 \setminus \Omega_2 = \{\mu \in \Omega_1 \mid \forall \mu' \in \Omega_2\}$$

where either μ and μ' are not compatible.

Definition 8.14 (Left Join). Let Ω_1 and Ω_2 be multisets of solution mappings and $expr$ be an expression. We define:

$$\Omega_1 \bowtie \Omega_2 = (\Omega_1 \bowtie \Omega_2) \cup (\Omega_1 \setminus \Omega_2)$$

Definition 8.15 (Union). Let Ω_1 and Ω_2 be multisets of solution mappings. We define:

$$\Omega_1 \cup \Omega_2 = \{\mu \mid \mu \in \Omega_1 \vee \mu \in \Omega_2\}$$

Definition 8.16 (Evaluation). Let D be an RDF dataset over T , t a triple pattern and P_1, P_2 graph patterns. Then the evaluation of a graph pattern over D , denoted by $[[\]]_D$, is defined recursively as follows:

$$[[t]]_D = \{\mu \mid dom(\mu) = var(t) \wedge \mu(t) \in D\}$$

$$[[(P_1 \text{ AND } P_2)]]_D = [[P_1]]_D \bowtie [[P_2]]_D$$

$$[[(P_1 \text{ OPT } P_2)]]_D = [[P_1]]_D \bowtie \setminus [[P_2]]_D$$

$$[[(P_1 \text{ UNION } P_2)]]_D = [[P_1]]_D \cup [[P_2]]_D$$

Graph Pattern	Solution Modifiers
BGP	ToList
Join	OrderBy
LeftJoin	Project
Filter	Distinct
Union	Reduced
Graph	Slice

TABLE 8.1: Graph Patterns & Solution Modifiers

$$[[[P \text{ FILTER } R]]]_D = \{\mu \in [[P]]_D \mid \mu \models R\}$$

where $var(t)$ is the set of variables occurring in t .

The next section details how the strategies described above are implemented in a query approach over RDF data.

A *SPARQL graph pattern* expression is defined as follows:

1. A tuple from $(IL \cup V) \times (I \cup V) \times (IL \cup V)$ is a graph pattern (a triple pattern).
2. If P_1 and P_2 are graph patterns, then expressions $(P_1 \text{ AND } P_2)$, $(P_1 \text{ OPT } P_2)$, and $(P_1 \text{ UNION } P_2)$ are graph patterns.
3. If P is a graph pattern and R is a SPARQL built-in condition, then the expression $(P \text{ FILTER } R)$ is a graph pattern.

A SPARQL built-in condition is built using elements of the set $V \cup IL$ and constants, logical connectives (\neg, \wedge, \vee), inequality symbols (e.g. $<, ! =, >$), the equality symbol ($=$), unary predicates (e.g. `bound`, `isBlank`, and `isIRI`).

The definitions above provide the main elements and abstraction behind the SPARQL query language. These elements are used both as: (i) a set of requirements on the expressivity that should be supported by schema-agnostic queries and (ii) key abstractions which are shared among the description of the proposed schema-agnostic query approach and SPARQL. The next sections introduce the elements of the proposed schema-agnostic query approach, including the *query analysis* and the *query processing* approaches.

8.5 Query Analysis

8.5.1 Motivation

The *query analysis* step consists of analyzing the schema-agnostic query into a set of *semantic features* and a *structured query representation* which supports *syntactic/structural* and *vocabulary approximation*. While the term ‘*Question Analysis*’ has been used in the context of QA, this work focuses on an abstraction which can be applied both to the *natural language scenario* (as in QA) but also that could be inherited by other schema-agnostic query scenarios, such as *structured schema-agnostic queries*, i.e. queries under a structured query syntax in which the user is abstracted from the database schema.

8.5.2 Query Representation

The query analysis process consists of the analysis of the schema-agnostic query input (in particular in the natural language query scenario), and the transformation of the original input query into a *structured query representation & feature set* which can be later used by the query processing approach. The structured query representation aims at providing a representation of the original query which is closer to the structured data model, also explicitly defining the core semantic query features present in the query. Differently from existing works in the NLI space, this work does not focus on providing a final SPARQL query output which is used as the basis for the query answering.

The query analysis outputs the following categories and abstractions:

- **Partial ordered dependency structure (PODS):** Consists of the set of entities connected by syntactic dependencies extracted from the schema-agnostic query in a graph format. The PODS representation targets mapping a lightweight syntactic structure which facilitates syntactic approximations.
- **Entity type patterns:** Consists of the typing of the core entities in the query according to their possible entity type (data model category): instances, classes, properties and complex classes and operators.
- **Functional Operators:** Maps the classification of terms related to database operations which are referred in the query. Possible operations are:
 - **Count**

- **Conditional operators:** Inequality/equality operators.
- **Ordering**
- **Ranking**
- **Logical Operators:** Maps to logical operators which are referred in the query.
 - AND
 - OR
- **Query Types:** Classifies the queries according to the possible data model elements types & operators present in the query.
 - *Queries with instances references*
 - *Queries with classes/complex classes references*
 - *Queries with operators references*
 - *Queries with constraint composition (path queries, conjunction & disjunction operators).*
- **Question Type:** Classification of the basic query types supported by the approach.
 - **Factoid:** “Who is the wife of Barack Obama?”.
 - **List:** “Give me all cities in the US with less than 10000 inhabitants.”.
 - **Definition:** “Who was Tom Jobim?”.
 - **Relationship:** “What is the connection between Barack Obama and Indonesia?”.
 - **Superlative:** “What is the highest mountain?”.
 - **Yes-No:** “Was Margaret Thatcher a chemist?”.
 - **Aggregation:** “How many states are there in the United States of America?”.
- **Answer Type:** The class of object sought by the question.
 - **Person:** (from “Who ”)

- **Place:** (from “Where ”)
- **Process & Method:** (from “How ”)
- **Date/Time:** (from “When ”)
- **Number:** (from “How many ”)
- **Question Focus:** Is the property or entity that is being sought by the question. Examples: “In which **city** was Barack Obama born?”, “What is the **population** of Galway?”.
- **Question Phrase:** Contains the part of the question that says what is being asked.
- Wh-words (“who”, “what”, “which”, “when”, “where”, “why”, and “how”)
- Wh-words + nouns, adjectives or adverbs: (“which party ...”, “which actress ...”, “how long ...”, “how tall ...”)

Despite targeting examples focusing on natural language queries, the categories and abstractions previously described can be generalized to other schema-agnostic query scenarios.

Most of the categories and abstractions described above are present in different NLI systems over structured and unstructured data. The combination of the *lightweight syntactic representation* (partial ordered dependency structure (PODS)), the *entity type patterns* and *query types* instead of a more rigid predicate-argument structure is a specific contribution of this work.

Along the chapter two queries, are used to demonstrate the approach:

- *Query example I:* ‘Who is the daughter of Bill Clinton married to?’
- *Query example II:* ‘What is the highest mountain?’

These queries were selected by their difference in terms of query processing strategy. Figure 8.2 and Figure 8.3 shows the set of query features associated with the two example queries used through this chapter.

In the next sections the steps involved in the *Query Analysis* are described.

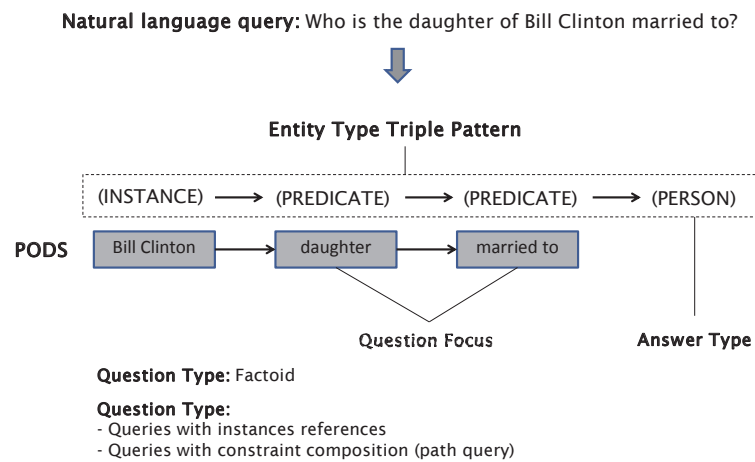


FIGURE 8.2: Example of the query analysis output for the query: ‘Who is the daughter of Bill Clinton married to?’.

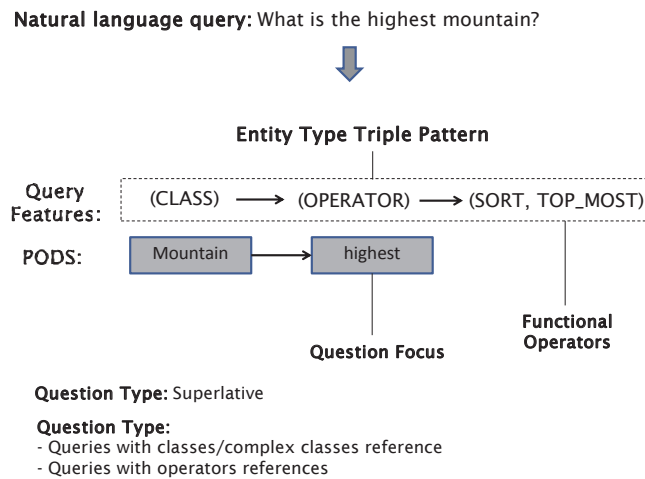


FIGURE 8.3: Example of the query analysis output for the query: ‘What is the highest mountain?’.

8.5.3 Query Analysis Steps

8.5.3.1 Overview

This section describes how the *structured query representation* & *feature set* is determined from the schema-agnostic query.

The query analysis workflow consists of the following components (Figure 8.4):

- *Query Parsing*
- *Entity Recognition & Classification*

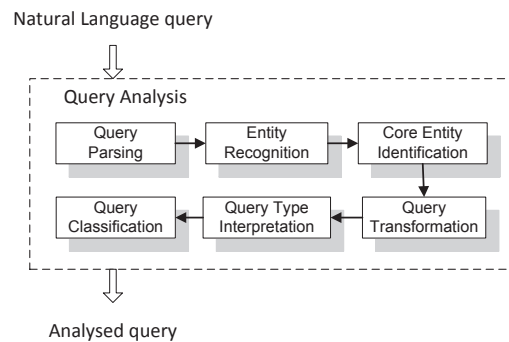


FIGURE 8.4: Components of the query analysis.

- *Core Entity Identification*
- *Query Transformation*

Each query analysis step is described in the following sections.

8.5.3.2 Query Parsing

The query analysis starts by parsing the natural language query. Two types of parsing are applied:

- **Part-of-speech (POS) Tagging:** POS tagging is the process of marking up a word in a text as corresponding to a particular *part-of-speech*(POS) (lexical or word category), based on its definition and its context, i.e. relationship with adjacent words in a phrase, sentence, or paragraph. This work uses POS Tags to detect compound nominals, to determine the classification of entity types. Particularly, this work uses the maximum entropy POS tagger described in [228, 229]. The accuracy of the tagger on the Penn Treebank is 96.86% overall and 86.91% on previously unseen words. An example of POS Tagging for the example query can be found in Figure 8.5.
- **Dependency parsing:** Dependency grammars can be traced back to the work of the Sanskrit grammarian Panini several centuries B.C. before the common era [230]. However, the work of Tesniere [231] is usually considered the starting point of the modern tradition of dependency grammars [230]. The basic assumption behind dependency grammars is that syntactic structures consist of lexical elements linked by binary asymmetrical relations called dependencies [230]. This implies the absence of phrasal nodes in the syntactic structure. According to [232] and [233] the core advantages of dependency grammars are associated to the fact that “*dependency links are close to the semantic*

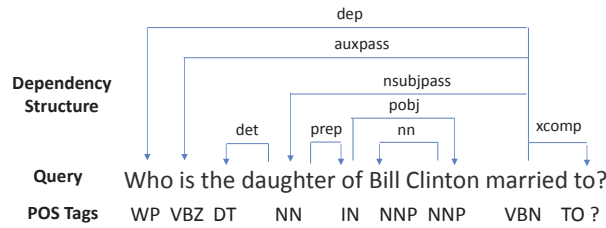


FIGURE 8.5: Example of the query POS Tagging and dependency parsing.

Who is the **[daughter]₁** of **[Bill Clinton]₂** **[married to]₃** ?
 What is the **[[highest]₁** **[mountain]₂**₃ ?

FIGURE 8.6: Example of the entity recognition for the query.

relationships needed for the next stage of interpretation.” [232] and “the dependency tree contains one node per word. Because the parsers job is only to connect existing nodes, not to postulate new ones, the task of parsing is in some sense more straightforward. [...]”

8.5.3.3 Entity Recognition & Classification

After the query is parsed, the entities in the query are recognized and classified according to the entity types that they probably match in the data: *instances*, *classes*, *properties*, *complex classes* and *operators*. Complex classes refer to the specification of classes with more than two words used as a descriptor (e.g. ‘HostCitiesOfTheSummerOlympicGames’). Additionally, terms mapping to functional and logical operations are also identified. While the lexical and structural variation for dataset elements is large, the vocabulary for typical database operations can be enumerated in a *lexicon of operations Op*.

The *entity recognition* step starts with the *question segmentation* step which uses rules which map part-of-speech and dependency relations into groups of words. This step consists of the phrasal segmentation (chunking) of the question string. This is done by the identification of the head nodes in the dependency structure and by aggregating their modifiers. The output for this step is a set of *entity candidate terms* which describe different possible combinations for the words in the query (different ways to segment the query). Figure 8.6 depicts the entity recognition output for the example query.

The entity classification step consists of a *rules-based classifier* which maps the *POS Tag patterns* into *entity types & operators* (instance, property, class, complex class, value,

Who is the **[daughter]₁** of **[Bill Clinton]₂** **[married to]₃** ?

Entity Types: **[daughter]₁** : (PREDICATE)
[Bill Clinton]₂ : (INSTANCE)
[married to]₃ : (PREDICATE)

What is the **[[highest]₁** **[mountain]₂**]₃ ?

Entity Types: **[highest]₁** : (OPERATOR) - (SORT, TOP_MOST)
[mountain]₂ : (CLASS)
[highest mountain]₃ : (CLASS)

FIGURE 8.7: Example of the entity classification for the query.

operator). The classifier takes into account POS Tags and adjacent dependency relations for the classification of the entity types. Operators and their associated parameters are also classified in this step. Figure 8.7 shows examples for the entity classification step.

8.5.3.4 Core Entity Identification

This step consists of the identification of the *core entity* in the query. The *core entity* is defined as the entity in the query which will be first aligned to the dataset, generating the *semantic pivot*. This step is related to the *semantic resolution ordering*, in which terms which are less bound to the ambiguity, vagueness, synonymy (AVS) conditions are resolved first and are used as a context mechanism to improve the probability of a correct semantic matching.

Two types of heuristics are used to determine the core entity:

POS-Tag based priority

Named entities define people, places, organisations, events, among others, and usually map to instances in RDF(S). Due to their lower propensity to the AVS conditions, proper nouns have the highest semantic matching priority in the query process.

Predication entities provide the description of sets (common/notable categories) and relations and map to classes, predicates and complex classes in RDF(S). The remaining lexical categories for predication entities are Nouns, Verbs, Adjectives, Adverbs and their combination.

The priority of lexical categories can be summarized as follows:

$$priority(ProperNoun) > priority(Noun) > priority(Verb) > priority(Adjective) > priority(Adverb).$$

Core Entity: [Bill Clinton] : (INSTANCE)
Core Entity: [mountain] : (CLASS)
Core Entity: [highest mountain]: (CLASS)

FIGURE 8.8: Determination of the core entities for the two example queries.

The priorities for multi-word expressions can be composed using the priorities of the independent lexical categories, e.g. $priority(Noun + Noun) > priority(Adjective + Noun) > priority(Adjective)$.

Figure 8.8 depicts two examples where for the first query a proper noun (*Bill Clinton*) is selected as the core entity following the above criteria. In the second example, two possible core entities candidates are selected: *mountain*, *highest mountain*.

Specificity-based priority

In case there are more than one core entity selected in the query, for example two disjoint nouns (e.g. ‘*Who were the astronauts which were women?*’) or proper nouns (e.g. ‘*Which films starred Julia Roberts and Richard Gere?*’), *specificity heuristics* are used to select the core entity. The core rationale behind the application of specificity-based measures is to determine how specific a term is in relation to a reference corpora. Following results derived from Zipf’s law [234], the number of senses associated with a word is correlated to its frequency in a reference corpus. The more specific a word, the smaller the number of contexts it occurs and less bound to the AVS conditions. In this work, *inverse document frequency* (IDF) is used as a heuristic specificity measure [191].

Definition 8.17 (Inverse Document Frequency (IDF)). Let n_{k_i} be the number of documents containing the term k_i and N the total number of documents in a reference corpora \mathcal{RC} . The *inverse document frequency* for the term k_i is given by:

$$idf_i = \log \frac{N}{n_{k_i}} \quad (8.1)$$

Definition 8.18 (Specificity Ordering). Let q be a query with its associated entities and predicates candidates, denoted by q_0, q_1, \dots, q_n . The query entities can be ordered into a sequence of query terms $\langle q'_0, q'_1, \dots, q'_n \rangle$ using a heuristic measure of specificity $h_{specificity}$ from the most specific to the less specific, that is, $\forall i \in [0, n], h_{specificity}(q'_i) \geq h_{specificity}(q'_{i+1})$, such that the query syntactic constraints are satisfied.

Term	IDF
neustadt	3.337
keynes	2.938
computing	2.025
concert	1.799
bridge	1.596
child	1.482
records	1.119
town	1.044

TABLE 8.2: Examples of IDF values a over Wikipedia 2013 corpus. The more specific words have higher IDF values.

8.5.3.5 Query Transformation

This step transforms the natural language query into a *lightweight syntactic structure* aiming at maximizing the vocabulary and syntactic matching between the query structure and the dataset structure. The dependency structures are used together with the POS Tags to define the *partial ordered dependency structures (PODS)*.

The query parsing module builds a PODS by taking as inputs dependency structures, the detected named entities and core entities, applying a set of transformation operations over the original Stanford dependencies. These operations reduce and re-order the original set of query dependencies. The core entity combined with the original dependency structure determine the ordering of the elements in the structure.

This transformation is done by applying three sets of operations: (i) removal of stopwords and their associated dependencies, (ii) merging of the dependency structures which a head node to its modifiers and (iii) re-ordering of the dependencies based on the core entity position in the query (where the core entity becomes the first query term and the topological relations given by the dependencies are preserved). Examples of PODS are depicted in Figure 8.9. Step (i) uses the output of the *entity recognition & classification* and step (iii) the *core entity identification* components.

Definition 8.19 (Partial Ordered Dependency Structure (PODS)). Let $T(V, E)$ be a *typed dependency structure* over the question Q where V and E are nodes and edges respectively. The *partial ordered dependency structure (PODS)* $D(V, E)$ of Q is defined by applying the following operations over T :

1. *merge*: adjacent nodes V_K and $V_{K+1} \in T$ where $E_{K,K+1} \in \{\text{nn, advmod, amod}\}$.
2. *eliminate*: the set of nodes V_K and edges $E_K \in T$ where $E_K \in \{\text{advcl, aux, auxpass, ccomp, complm, det}\}$.
3. *replicate*: the triples where $E_K \in \{\text{cc, conj, preconj}\}$.

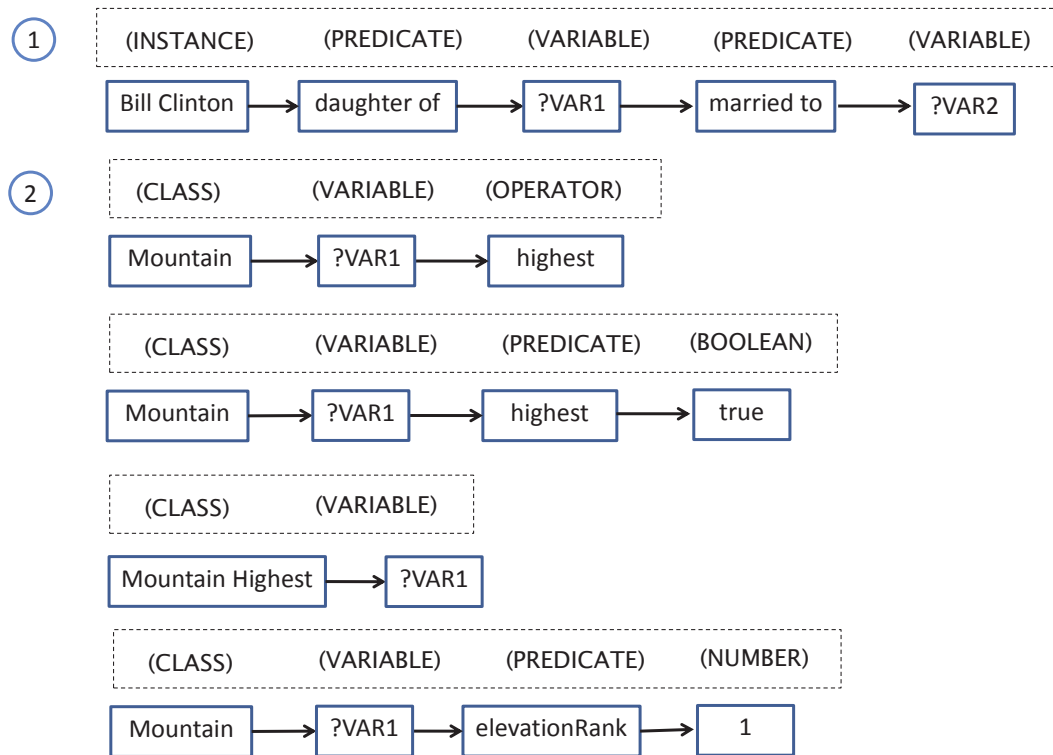


FIGURE 8.9: Possible interpretation for the query examples.

where the edge labels *advmod*, *amod*, ... represent the specific dependency relations (see [233] for a the complete list of dependencies). In the definition above, the *merge* operation consists of collapsing adjacent nodes into a single node for the purpose of merging head-modifiers into a single node, in complement with the entity recognition output. The *eliminate* operation is defined by the pruning of a node-edge pair and eliminates concepts which are not semantically relevant or covered in the representation of data in RDF(S). The *replicate* operation consists of copying the remaining elements in the PODS for each coordination or conjunctive construction.

8.5.4 Query Entity Types & Database Structural Interpretation

The *query entity types* consist of the list of possible data model types associated with the PODS entities. The *database syntactic interpretation* consists of the possible mappings to basic graph patterns, defining an explicit typing of the data model types and functional/logical operators.

The database structural interpretation only takes into account the possible association between words and data elements types (classes, instances, properties, etc) and the

Triple type pattern
INSTANCE-PREDICATE-VARIABLE
VARIABLE-PREDICATE-INSTANCE
VARIABLE-PREDICATE-VARIABLE
CLASS-TYPE-VARIABLE
VARIABLE-TYPE-CLASS
COMPLEX_CLASS-TYPE-VARIABLE
VARIABLE-TYPE-COMPLEX_CLASS
INSTANCE-PREDICATE-INSTANCE
VARIABLE-PREDICATE-VALUE
INSTANCE-PREDICATE-VALUE

TABLE 8.3: Basic type graph patterns.

Operator
AGGREGATE
ORDER
COMPARISON
DISJUNCTION
CONJUNCTION

TABLE 8.4: Types of operators.

associated structural interpretations, not covering the lexical and abstraction-level variations, which are resolved at the query processing step. Tables 8.3 and 8.4 list the set of *basic type graph patterns*. Compositions of basic type graph patterns and operators define the query constraints.

The database syntactic interpretation process maps dependency and POS Tag patterns into the interpretation patterns using a rules-based method. Since the set of basic type graph patterns is a small enumerable set (Table 8.3), the number of mapping patterns is enumerable. One query can have one or more associated syntactic query interpretations.

8.5.5 Query Classification

The final step consists of the classification of the query according to a set of *query features* which play a central role in the definition of the *query plan* algorithm (Section 8.6.4). Each query is classified according to one or more query features. The set of query features are listed below and are based on references to entity types, composition patterns and operator types present in the query:

1. *Queries with instance references*
2. *Queries with classes/complex classes references*
3. *Queries with operator references*

4. *Queries with constraint composition*

- *Path queries*
- *Conjunction & Disjunction operators*

The set of features map to database primitives in the possible *data model type* of the *associated semantic pivot* (instance, class, complex class), *operators* (e.g. comparative, ordering operators) and *structural/compositional patterns* (conjunction, disjunction, property path). The query features for the example queries are shown in Figure 8.2 and Figure 8.3.

8.6 Schema-agnostic Query Processing

8.6.1 Overview

The *query processing* approach consists of the composition of a set of *semantic search*, *graph navigation* and *solution modifier* operations over the $\tau - Space$ and over the *data graph structure*. The sequence of operations $\langle op_0 \dots op_n \rangle$ for a query Q defines its *query processing plan*. The query plan is built by taking into account the structured query representation out of the *query analysis*.

The operations are described in the following subsections.

8.6.2 Search operations

Search operations map PODS query terms to the elements in the graph G associated to the database DB . They consist of *distributional* and *term search operations* over the graph G embedded in the distributional VS^{dist} and term VS^{word} reference frames and over the subspaces associated with each data model category within the $\tau - Space$. The distributional semantic search operations use the distributional semantic search approach defined in Chapter 7, with the *distributional semantic relatedness measure as a ranking function* and the application of the *semantic differential principles* in the determination of the *semantic threshold*.

In all search operations, all the URIs are assumed to have meaningful natural language descriptors associated, i.e. descriptors which are composed by words in a language which is shared by the reference corpora and by the external agent querying the system.

8.6.2.1 Instance search

As previously discussed, the *instance search* model prioritizes term-based matching, due to the lower variability in the naming of instances (less impact from the AVS conditions) and to the potential higher dimensionality of the distributional vector space (due to the proportionally higher number of instances in the DB).

The instance search operation consists of mapping the terms associated to *query instance candidates* $m(q^I)$ in the analysed query, to the instances in G , i.e. $m(q^I, i), \forall q^I \in Q, \forall i \in I$.

The *primary instance search* consists of a keyword search over the *space of instance terms* $VS^{I\{word\}}$. For a query term q^I over the term space $VS^{I\{word\}}$, the *ranking function* $\phi(\mathbf{q}, \mathbf{i}), \forall i \in I$ is given by a combination of:

- *tf/idf of instance terms in the database*: $s_{tf/idf}(q^I, i)$.
- *dice coefficient*: between the instance terms and the query term $sim_{dice}(q^I, i)$. Prioritizes closer string matching between query terms and instance labels. The main purpose of this function is to remove terms with a partial matching ($q^I = \text{'Barack Obama'}$ with 'Michelle Obama') and also where q^I has a full matching with a label with a larger string ($q^I = \text{'Barack Obama'}$ with $\text{'Barack Obama Sr.'}$ or $\text{'Barack Obama Office'}$).
- *node cardinality*: number of tuples (triples) $n(i)$ associated with i .

$s_{tf/idf}(q^I, i)$ and $sim_{dice}(q^I, i)$ are used as filters: instances with values below a threshold are filtered. $sim_{dice}(q^I, i)$ and $n(i)$ are used as ranking functions. The instance search returns a list of URIs associated with the instances i which has at least one associated matching word to q^I . For all the instances containing at least one of the keywords associated with the query, the list is ranked according to string similarity. The ranking policy based on the node cardinality states that for homonymous instances, more popular instances are prioritized.

The ranking algorithm for the instance search is given by Algorithm 4.

In Algorithm 4, $rank(R_I, \phi_{dice}, \phi_n)$ is a function which ranks the set of returned instances, according to the dice coefficient and the number of triples associated to the matching instance.

The secondary instance search step consists of the distributional semantic search over the space of instance context vectors $VS^{I\{dist\}}$ (Algorithm 5). The distributional instance

Algorithm 4 Instance term search: $\zeta_{VSI\{word\}}(\mathbf{q}^I, \mathbf{i})$

I : set of instances in DB .
 q^I : query term candidate for mapping to instance.
 R_I : set of matched instances to the query term q^I .
 $VSI\{word\}$: instance term space.
 $\eta_{tf/idf}, \eta_{dice}$: thresholds.
for all $i \in I$ **do**
 $\phi_{tf/idf} \leftarrow s_{VSI\{word\}}(\mathbf{q}^I, \mathbf{i})$
 if $\phi_{tf/idf} \geq \eta_{tf/idf}$ **then**
 $\phi_{dice} \leftarrow sim_{dice}(q^I, i)$
 if $\phi_{dice} \geq \eta_{dice}$ **then**
 $(r_I, \phi_n) \leftarrow n(i)$
 $R_I \leftarrow r_I$
 end if
 end if
end for
 $R_I \leftarrow rank(R_I, \phi_{dice}, \phi_n)$

search operation is defined by the computation of the semantic relatedness measure s between the corresponding distributional vector of the instance candidate term $\vec{\mathbf{q}}^I$ and the instances in $VSI\{dist\}$.

Algorithm 5 Instance distributional search: $\zeta_{VSI\{dist\}}(\mathbf{q}^I, \mathbf{I}), \eta_{I\{dist\}}$

I : set of instances in DB .
 q^I : candidate query term for instance.
 R_I : set of matched instances to the query q^I .
 $VSI\{dist\}$: instance distributional space.
 $\eta_{I\{dist\}}$: threshold.
for all $i \in I$ **do**
 $\phi_{dist} \leftarrow s_{VSI\{dist\}}(\vec{\mathbf{q}}^I, \vec{\mathbf{i}})$
 if $\phi_{dist} \geq \eta_{I\{dist\}}$ **then**
 $R_I \leftarrow r_I$
 end if
end for
 $R_I \leftarrow rank(R_I, \phi_{dist})$

In Algorithm 5, $rank(R_I, \phi_{dist})$ is a ranking function which ranks the matching instances according to the distributional semantic relatedness value ϕ_{dist} with regard to a reference corpora RC .

8.6.2.2 Class search

Classes are more bound to vagueness, ambiguity and synonym, being more sensitive to vocabulary variation. The *class search operation* is defined by the computation of the semantic relatedness measure s between the class candidate term q^C and the class entities

in the $VS^{C\{dist\}}$ ($s_{VS^{C\{dist\}}}(\vec{q}^C, \vec{c})$, $\forall c \in C$). Algorithm 6 describes the procedure for class search.

Algorithm 6 Distributional class search (non-contextualised): $\zeta_{VS^{C\{dist\}}}(\vec{q}^C, \vec{c}, \eta_{C\{dist\}})$

C : set of classes in DB .
 q^C : candidate query term for class.
 R_C : set of matched classes to the query q^C .
 $VS^{C\{dist\}}$: class distributional space.
 $\eta_{C\{dist\}}$: threshold.
for all $c \in C$ **do**
 $\phi_{dist} \leftarrow s_{VS^{C\{dist\}}}(\vec{q}^C, \vec{c})$
 if $\phi_{dist} \geq \eta_{C\{dist\}}$ **then**
 $R_C \leftarrow r_C$
 end if
end for
 $R_C \leftarrow rank(R_C, \phi_{dist})$

In Algorithm 6, $rank(R_C, \phi_{dist})$ is a ranking function which ranks the matching classes according to the distributional semantic relatedness value ϕ_{dist} with regard to a reference corpora R_C . The search over the non-contextualised class space ($VS^{C\{dist\}}(c)$) is used when the class is the semantic pivot.

When a set of instances i are the semantic pivot, the contextualised class search is defined on the subspace associated with ($i VS^{C\{dist\}}(i)$).

Algorithm 7 Distributional class search (contextualised): $\zeta_{VS^{C\{dist\}}(i)}(\vec{q}^C, \vec{c}, \eta_{C\{dist\}})$

i : instance i .
 C^i : set of classes associated with an instance i in DB .
 q^C : candidate query term for class.
 R_C : set of matched classes to the query q^C .
 $VS^{C\{dist\}}(i)$: distributional class subspace associated instance i .
 $\eta_{C\{dist\}}$: threshold.
for all $c \in C$ **do**
 $\phi_{dist} \leftarrow s_{VS^{C\{dist\}}(i)}(\vec{q}^C, \vec{c})$
 if $\phi_{dist} \geq \eta_{C\{dist\}}$ **then**
 $R_C \leftarrow r_C$
 end if
end for
 $R_C \leftarrow rank(R_C, \phi_{dist})$

The output of the search is a ranked list of URIs which are semantically related to q^C . The list of URIs is ranked by their semantic relatedness score in relation to q^C , according to a reference corpus R_C .

8.6.2.3 Property search ($VS^{P\{dist\}}(i)$ & $VS^{P\{dist\}}$)

The *property search* consists of the semantic search of properties associated within the context of a set of instances which are directly or indirectly referred to in the query. The set of instances define subspaces of associated property entities in the $\tau - Space$ which define the target search subspace. Due to the dimensionality of the distributional space $VS^{P\{dist\}}(i)$ the definition of the instance pivots support a dimensional reduction during the search process which is based on the context defined by the instances referred to in the query.

The query term q^P is used as an input for a distributional semantic search over the properties associated with the instance subspace. The search is defined by $s_{VS^{P\{dist\}}(i)}(\vec{\mathbf{q}}^P, \vec{\mathbf{p}})$, $\forall p \in P$.

Algorithm 8 Distributional property search (contextualised):

$\zeta_{VS^{P\{dist\}}(i)}(\vec{\mathbf{q}}^P, \vec{\mathbf{p}}(\mathbf{i}), \eta_{P\{dist\}})$

i : instance i .

P^i : set of properties associated with an instance i in DB .

q^P : candidate query term for property.

R_P : set of matched property to the query q^P .

$VS^{P\{dist\}}(i)$: distributional property subspace associated instance i .

$\eta_{P\{dist\}}$: threshold.

for all $p \in P$ **do**

$\phi_{dist} \leftarrow s_{VS^{P\{dist\}}(i)}(\vec{\mathbf{q}}^P, \vec{\mathbf{p}})$

if $\phi_{dist} \geq \eta_{P\{dist\}}$ **then**

$R_P \leftarrow r_P$

end if

end for

$R_P \leftarrow rank(R_P, \phi_{dist})$

The search of the complete property space (non-contextualised property search) $VS^{P\{dist\}}$ can be performed when there is no instance semantic pivot defined:

8.6.3 Constraint Composition & Solution Modifiers

8.6.3.1 Constraint Composition

Constraint composition are operations in which attribute constraints (basic graph patterns) are composed into complex graph patterns using different composition patterns. There are three main types of constraint composition operations:

Algorithm 9 Distributional property search (non-contextualised) :

$\zeta_{VSP\{dist\}}(\vec{q}^P, \vec{p}, \eta_{P\{dist\}})$

P : set of properties in DB .
 q^P : candidate query term for property.
 R_P : set of matched property to the query q^P .
 $VSP\{dist\}$: distributional property subspace.
 $\eta_{P\{dist\}}$: threshold.
for all $p \in P$ **do**
 $\phi_{dist} \leftarrow s_{VSP\{dist\}}(\vec{q}^P, \vec{p})$
 if $\phi_{dist} \geq \eta_{P\{dist\}}$ **then**
 $R_P \leftarrow r_P$
 end if
end for
 $R_P \leftarrow rank(R_P, \phi_{dist})$

Property path composition: Consists of a predicate composition that defines a *path query*. The semantic property composition is determined for a path of properties connected through a common instance. This operation maps to the following graph pattern: $p_0(i, v_0) \wedge_{n=0}^N p_{n+1}(v_n, v_{n+1})$, where v_i represents a variable and p_n represents the set of predicates such that for all $q^P \in Q^P$, $s(\vec{q}^P, \vec{p}) > \eta_P$.

The path composition is defined by the following navigation function:

Algorithm 10 Property path composition resolution algorithm.

i : set of pivot instances.
 Q^P : sequence of query properties.
 p_r : ordered list of properties.

for all $q^P \in Q^P$ **do**
 $P \leftarrow \bigcup_{p \in P} p(i)$
 $R_P \leftarrow \zeta(q^P, P, \eta_P)$
 for all $r_P \in R_P$ **do**
 $i_{to} \leftarrow navigateTo(r_P)$
 \leftarrow r_P(i, i_{to})
 $i \leftarrow i_{to}$
 end for
end for

Geometrically, the property composition is defined by a sequence of translations over VSP^{dist} (Figure 8.10).

Extensional class expansion (instance listing for a class): $(\xi(i))$ Consists of expanding the set of instances I associated with a class c through *rdf:type*. This operation maps to the $type(c, v_n)$ triple pattern, where v_n defines a set of instances $\in I$ associated with the class c . The extensional expansion can be generalized to include the definition of sets using properties.

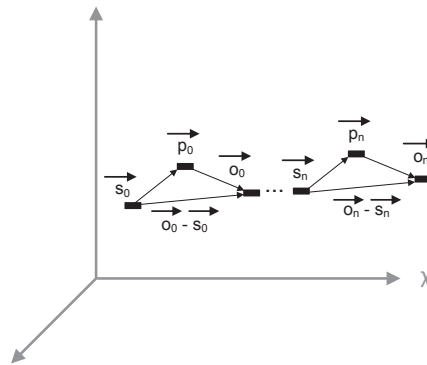


FIGURE 8.10: Vector representation for the property path composition.

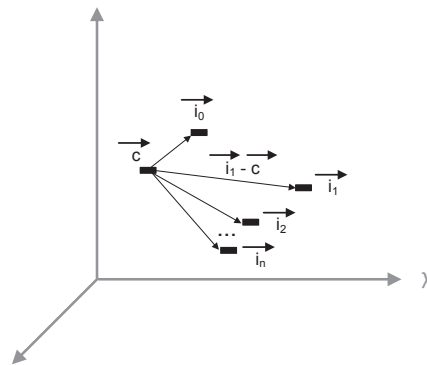


FIGURE 8.11: Vector representation for the extensional expansion.

Geometrically, the extensional class expansion consists of a set of translation vectors over VS^{dist} which have the same origin in a class element (Figure 8.11).

Star-Shaped property composition: Consists of the composition of triple patterns in a *disjunctive*: $\bigwedge_{n=0}^N p_n(term, v_n)$, where v_i represents a variable or *conjunctive* form: $\bigvee_{n=0}^N p_n(term, v_n)$, where $term \in I \cup C \cup V$

Geometrically, the star-shaped property composition consists of a set of translation vectors over VS^{dist} which have the same origin (Figure 8.12).

8.6.3.2 Solution Modifiers

Solution modifiers consists of the set of operations for filtering triples ($f : T \rightarrow T$) or mapping triples to the real domain ($f : T \rightarrow \mathbb{R}$). The following definitions are based on [224].

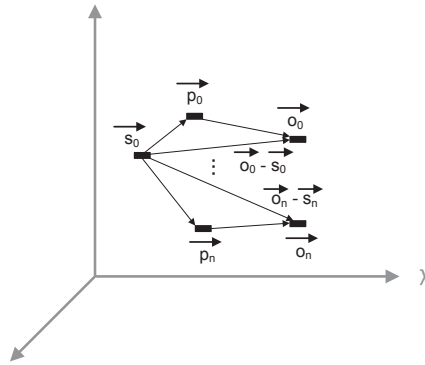


FIGURE 8.12: Vector representation for the star-shaped property composition.

Definition 8.20 (Solution Sequence Modifiers). A solution sequence modifier consists of one of the following operations:

- **List:** is used where conversion from the results of graph pattern matching to sequences occurs.
- **Order By:** order the solution.
- **Projection:** select certain output variables.
- **Slice:** provides a subset of the solution tuples (as the combination of OFFSET and LIMIT).
- **Exists(Yes/No):** defines if a proposition is based on the *DB*.

Definition 8.21 (List). Let Ω be a multiset of solution mappings. We define:

$List(\Omega_1) =$ a sequence of mappings $\mu \in \Omega$ in any order

Definition 8.22 (Order By). Let Ψ be a sequence of solution mappings. We define:

$OrderBy(\Psi, condition) = \{ \mu \mid \mu \in \Psi \text{ and the sequence satisfies the ordering condition} \}$

where *condition* can be *ascending*, *descending* according to a numerical or lexicographical criteria.

Definition 8.23 (Projection). The projection operator (π) restricts a relation to a subset of its attributes. Let Ω be a sequence of solution mappings and V a set of variables. For mapping μ , write $Proj(\mu, V)$ to be the restriction of μ to variables in V .

$$\pi(\Psi, V) = \{ \pi(\Psi[\mu] V) \mid \mu \in \Omega \}$$

Definition 8.24 (Constant Projection). $\pi_{p_0, \dots, p_i}(c)$ is the *constant projection*, which defines the set of predicates p_i assigned to a constant c .

Definition 8.25 (Slice). Let Ψ be a sequence of solution mappings. We define:

$$\text{Slice}(\Psi, \text{start}, \text{length})[i] = \Psi[\text{start} + i], \text{ for } i = 0 \text{ to } (\text{length}-1)$$

Definition 8.26 (Exists(Yes/No)). *exists* is a function that returns true (yes) if the pattern evaluates to a non-empty solution sequence; otherwise it returns false (no).

Definition 8.27 (Aggregation modifiers). Maps a set of triples or entities from VS^{dist} or VS^{word} into the \mathbb{R} domain ($f : T, V \rightarrow \mathbb{R}$), based on an enumerable set of functional operators Op (e.g. Count, Sum, etc).

– **Count:** Count is a set function $Count : \Omega \rightarrow \mathbb{I}$ which counts the number of a given condition appears. $Count(\Omega) = \text{card}[N]$.

– **Sum:** Sum is a set function $Sum : \Omega \rightarrow \mathbb{R}$ which sums the values within a multiset. $Sum(\Omega) = Sum(List(\Omega))$.

$$\begin{aligned} Sum(S) &= S_1 + Sum(S_2 \dots n), \text{ when } \text{card}[S] > 1 \\ Sum(S) &= S_1 + 0, \text{ when } \text{card}[S] = 1 \\ Sum(S) &= 0, \text{ when } \text{card}[S] = 0 \end{aligned}$$

– **Avg:** The Avg set function $Avg : \Omega \rightarrow \mathbb{R}$ calculates the average value for an expression over a multiset Ω . $Avg(\Omega) = 0$, where $Count(\Omega) = 0$ $Avg(\Omega) = Sum(\Omega)/Count(\Omega)$, where $Count(\Omega) > 0$

– **Top:** Top is a set function $Top : \Omega \rightarrow \mathbb{R}$ that return the maximum value from a multiset Ω .

$$Top(\Omega) = \text{Slice}(\text{OrderBy}(\Omega, 'descending'), 0, 1)$$

– **Bottom:** Bottom is a set function $Bottom : \Omega \rightarrow \mathbb{R}$ that return the minimum value from a multiset Ω .

$$Bottom(\Omega) = \text{Slice}(\text{OrderBy}(\Omega, 'ascending'), 0, 1)$$

– **Sample:** Sample is a set function $Sample : \Psi \rightarrow RDFTerm$ which returns an arbitrary element from the multiset Ω .

$Sample = \text{Slice}(\Psi, \text{Random}(0, \text{length}), \text{length})$, where $\text{Random}(x, y)$ generates a number between x and y .

Definition 8.28 (Operation Lexicon). The *operation lexicon* Lex_{Op} contains the labels for *aggregation modifiers*.

8.6.3.3 User Feedback Modifiers

User feedback modifiers are functions which filter a set of triples based on the user input of a set of instances, classes and predicates. These operations aim at allowing users to cope with possible errors in the term and distributional search operations over the $\tau - Space$, by allowing them to select from a list matching the search criteria, a set of valid instances, classes and properties. The user feedback dialogs target just a filtering function, where users can select from a reduced list of options (maximum 5 elements) in case there is ambiguity in the term/distributional search process. The feedback function ($\psi : \Psi \rightarrow \Psi$) is defined as:

$$\psi(\Psi) = \bigcup_{n=0}^{length} Slice(\Psi, DialogSelect(\Psi, n), length)$$

where $\psi(\Psi)$ returns the index of $\mu \in \Psi$ selected by the user.

8.6.4 Operation Composition & Planning

All the *search*, *constraint composition* & *solution modifier* operations are organized into a *query planning algorithm* (Algorithm 11) which orchestrates the operations defined above, taking as an input the PODS.

The query processing algorithm works as a *semantic best-effort* query approach, where the algorithm maximizes the amount of semantic constraints which are matched, but eventually can return approximate results. The search operations and the constraints application are done in a semantic structured inverted index, aiming at performance and scalability). In the proposed query processing approach, instead of returning a structured query with a rigid syntax and rigid symbols, the approach can be interpreted as navigating through the data graph doing both semantic and syntactic approximations.

8.6.5 Geometrical Interpretation

The query processing approach consists of a set of *semantic search*, *constraint composition* and *solution modifier operations* over the $\tau - Space$ and over the data graph. The semantic search operations are performed over the different subspaces of the $\tau - Space$ model and have an associated *geometrical interpretation*.

The query sequence is embedded in the vector spaces VS^{dist} , VS^{word} , allowing to identify it with the a sequence of vectors $\langle \vec{q}'_0, \vec{q}'_1, \dots, \vec{q}'_n \rangle$.

In the first iteration, $\vec{q}'_0 \in VS^{dist}$, the vector representation of the semantic pivot q'_0 can be resolved to a vector \vec{e}_0 . If the entity e_0 is an instance, the associated predication

Algorithm 11 Distributional query planning algorithm

```

Q(VQ, EQ, OpQ) : Partial ordered dependency structure (PODS).
G(VG, EG) : RDF(S) graph.
A(VA, EA, P) : answer graph.
i : set of instances URIs.
c : set of classes URIs.
p : related properties URIs.
initialize(A)
q : query term
for all q ∈ VQ do
  if (isCoreEntity(q)) then
    i ← searchInstances(q)
    c ← searchClasses(q)
  end if
  if (isAmbiguous(i, c)) then
    i, c ← disambiguatePivotEntity(i, c)
  end if
  if (pivotEntityIsClass) then
    i ← extensionalExpansion(c)
  end if
  p ← searchProperties(i, q)
  if (hasOperations(Q)) then
    p ← searchOperations(i, Op)
  end if
  if (isAmbiguous(p)) then
    p ← disambiguateProperty(i, c)
    triples ← selectByPivotAndProperty(i, p)
  end if
  i, c ← navigateTo(triples)
  VA, EA ← triples
  PA ← applyOperation(triples, Op)
end for

```

subspace (which spans the relations and predications associated with the entity e_0) can be used to do the semantic approximation of the next query term. The second query term q'_1 can be matched with one or more relations and attributes associated with e_0 , for example p_0 , considering that $s(\vec{q}'_1, \vec{p}_0) \geq \eta_p$, where η_p is a semantic relatedness threshold. The entities associated with p_0 (for example e_1) are used as new semantic pivots.

At each iteration of the querying process, a set of semantic pivots are defined and are used to navigate to other points in the VS^{dist} . This navigation corresponds to the reconciliation process between the query and the entity dataset G . The reconciliation process can be defined as the sequence of vectors $\langle (\vec{q}'_1 - \vec{p}_1), (\vec{q}'_2 - \vec{p}_2), \dots, (\vec{q}'_n - \vec{p}_n) \rangle$. The proposed approximate querying process can also be represented geometrically as the vectors $\langle (\vec{e}_0 - \vec{p}_0), (\vec{p}_0 - \vec{e}_1), \dots, (\vec{p}_{n-1} - \vec{e}_n) \rangle$ over the τ -Space, which geometrically represents the process of finding the answer in the graph.

8.6.6 Query Processing Examples

In this section, the *query planning algorithm* is executed for the example queries.

8.6.6.1 Query Example I:

For the example query ‘*Who is the daughter of Bill Clinton married to?*’, the set of query processing steps are described for the model parameters below:

- Schema-agnostic query (different predicates from the database) Q : $\text{daughter}(\text{Bill Clinton}, ?x_0) \wedge \text{married to}(?x_0, ?x_1)$
- \mathcal{DB} : (DBpedia) $\text{childOf}(\text{Bill Clinton}, \text{Chelsea Clinton}), \text{spouse}(\text{Chelsea Clinton}, \text{Marc Mezvinsky}), \dots$
- \mathcal{DSM} : ESA (\mathcal{C} = document, \mathcal{W} = TF/IDF)
- \mathcal{RC} : Wikipedia 2013
- $\eta_p = 0.02, \eta_i = 0.9$ (more restrictive threshold for constants)

With the PODS and the query features, the query processing approach starts by resolving the *core (pivot) entity* in the query (in this case *Bill Clinton*) to the corresponding database entity (*dbpedia: Bill_Clinton*) (Figure 8.13).

After *Bill Clinton* is resolved, the subspace of the entity *dbpedia:Bill_Clinton* is selected, constraining the search space to elements associated with *dbpedia:Bill_Clinton*, and the next term in the PODS (‘*daughter*’) is used as a query term for a distributional semantic search over the neighboring elements of *dbpedia:Bill_Clinton*. The distributional semantic search is equivalent to computing the *distributional semantic relatedness* between the query term (‘*daughter*’) and all predicates associated with *dbpedia:Bill_Clinton* (*dbprop:religion, dbprop:child, dbprop:almaMater*, etc). The semantic equivalence between ‘*daughter*’ and *dbprop:child* is determined by using the corpus-based distributional commonsense information (the words ‘*daughter*’ and ‘*child*’ occur in similar contexts). A *threshold* filters out unrelated relations. After the alignment between ‘*daughter*’ and *dbprop:child* is done, the query processing *navigates to* the entity associated with the *dbprop:child* relation (*dbpedia:Chelsea_Clinton*) and the next query term (‘*married*’) is taken. At this point the entity *dbpedia:Chelsea_Clinton* defines the search subspace (relations associated with *dbpedia:Chelsea_Clinton*) and the semantic search for predicates which are semantically related to ‘*married*’ is done. The query

term ‘*married*’ is aligned to *dbprop:spouse* and the answer to the query is found: the entity *dbpedia:Mark_Mezvinsky* (Figure 8.14).

The same query process can be described in a more formalized way by using notation based on relational algebra.

The query process starts with the selection of the first query element which will be aligned to the dataset (the semantic pivot). In the example query, ‘*Bill Clinton*’ is the semantic pivot ($\{\text{Bill Clinton}\} = \gamma(Q)$). Since the semantic pivot is an instance, it is compared with the set of instances in the $VS^I\{\text{word}\}$ using the semantic threshold η_i :

$$\eta_i: \{:\text{Bill Clinton}\} \leftarrow \bigcup_{\forall i \in I} \zeta(\text{‘Bill Clinton’, } i, \eta_i)$$

The context is defined by the first alignment $\kappa_{\mathcal{DB}} = \{:\text{Bill Clinton}\}$. The next step consists of the application of the *projection operator* to get the set of predicates associated with the instance *:Bill Clinton*. $\pi_{\{\text{childOf}, \text{occupation}, \dots, \text{almaMater}\}}(\text{:Bill Clinton})$ (Figure 8.13). The algorithm then gets the query predicate associated with Bill Clinton, defining it as a query context $\kappa_Q = \{\text{daughter}\}$. It then computes the *semantic relatedness* between κ_Q and the predicates in the projection $\pi_{\{\text{childOf}, \text{occupation}, \dots, \text{almaMater}\}}(\text{Bill Clinton})$, selecting:

$$\{\text{childOf}\} \leftarrow \bigcup_{\forall p \in \pi_{\{\text{childOf}, \dots, \text{almaMater}\}}} \zeta(\text{‘daughter’, } p, \eta_p),$$

defining the first predicate substitution $\lambda_{\text{daughter}/\text{childOf}}$.

It then follows with the selection associated to the first part of the query $\{\text{Chelsea Clinton}\} \leftarrow \sigma_{\text{childOf}(\text{Bill Clinton}, ?x)}$ which redefines $\kappa_{\mathcal{DB}} = \{\text{Chelsea Clinton}\}$ followed by the *projection operator* $\pi_{\{\text{religion}, \text{occupation}, \dots, \text{spouse}\}}(\text{Chelsea Clinton})$. The query predicate associated with Chelsea Clinton is selected ($\kappa_Q = \{\text{married to}\}$).

$$\{\text{spouse}\} \leftarrow \bigcup_{\forall p \in \pi_{\{\text{religion}, \text{occupation}, \dots, \text{spouse}\}}} \zeta(\text{‘married to’, } p, \eta_p)$$

and the semantic relatedness is computed, defining the second predicate alignment $\lambda_{\text{marriedto}/\text{spouse}}$.

$$\{\text{Marc Mezvinsky}\} \leftarrow \sigma_{\{\text{childOf}(\text{Bill Clinton}, x_0) \wedge \text{spouse}(x_0, ?x_1)\}}$$

A query plan maps to multiple operations over the index. Figure 8.13 and Figure 8.14 depicts the steps for answering the query over the graph G associated with \mathcal{DB} .

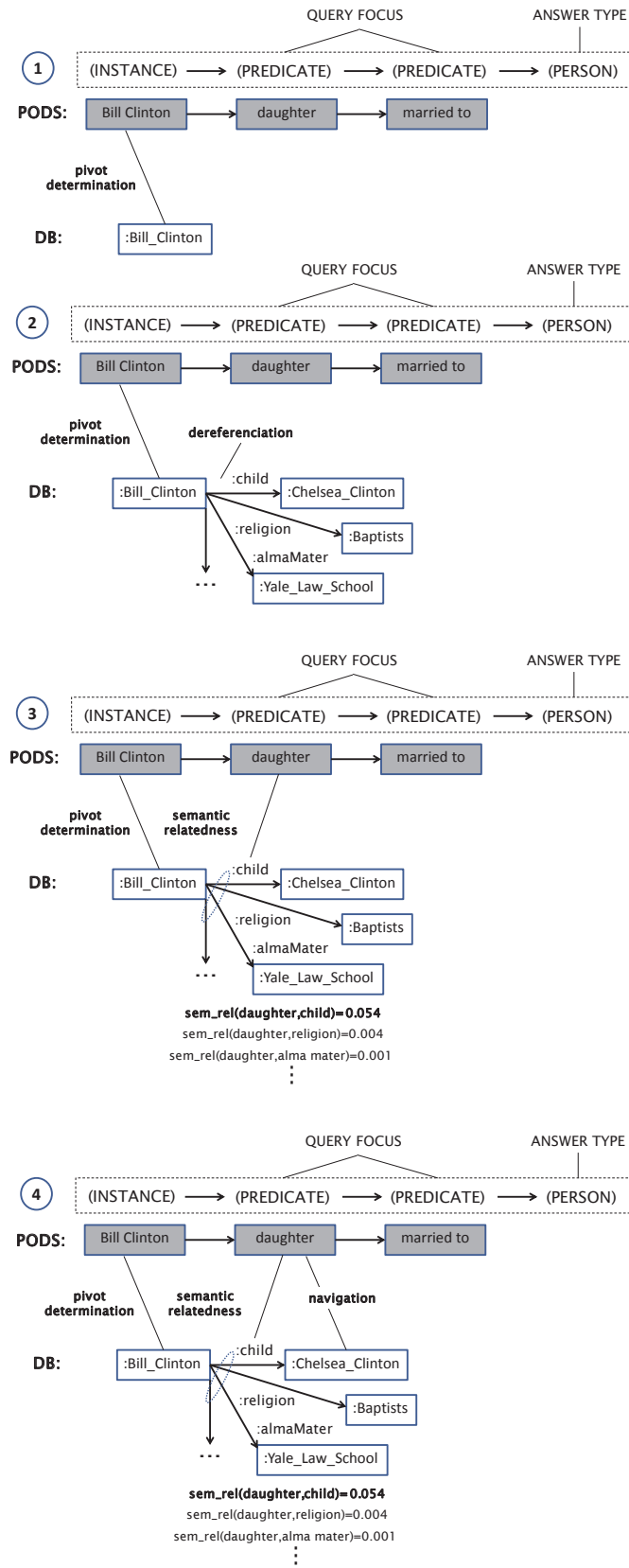


FIGURE 8.13: Query processing steps for the query example (Part I).

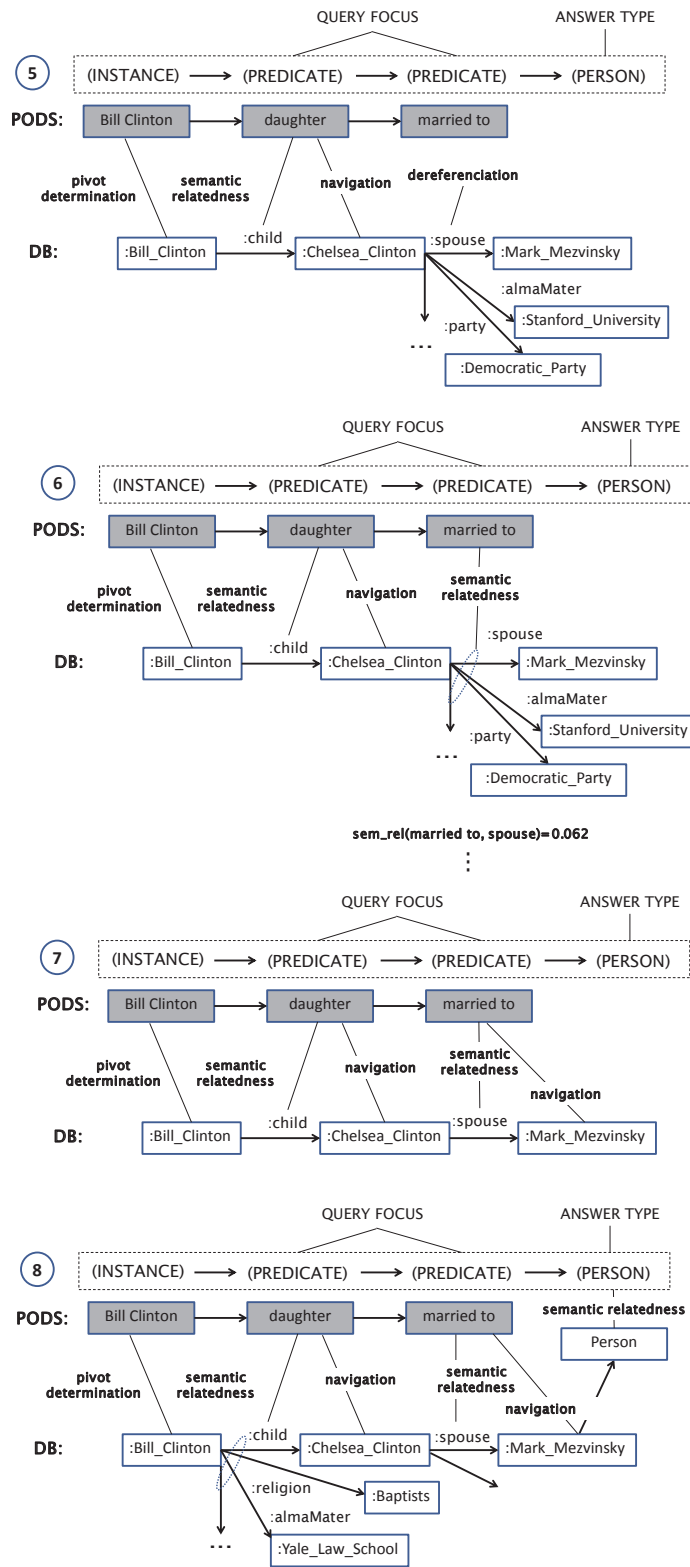
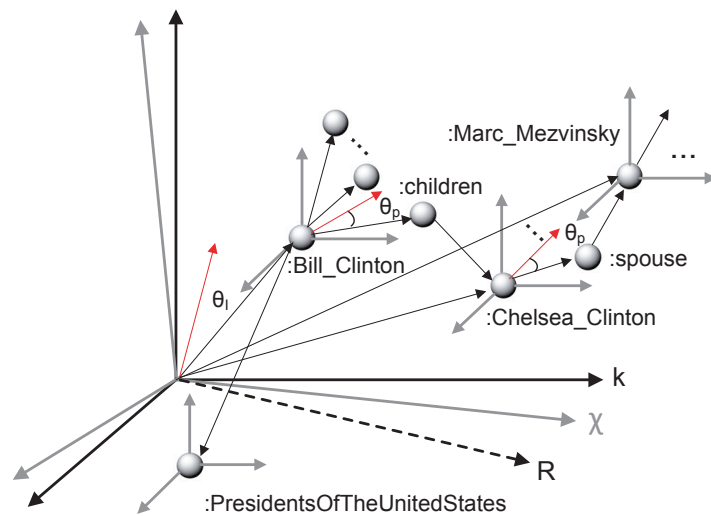


FIGURE 8.14: Query processing steps for the query example (Part II).

FIGURE 8.15: τ - Space query.

8.6.6.2 Query Example II:

For the example query ‘*What is the highest mountain?*’, the set of query processing steps are described for the model parameters below:

- Schema-agnostic query (different predicates from the database) Q : $\text{Mountain}(?x_0) \wedge \text{highest}(?x_0)$
- \mathcal{DB} : (DBpedia) $\text{type}(\text{Mount Everest}, \text{Mountain}), \dots$
- \mathcal{DSM} : ESA (\mathcal{C} = document, \mathcal{W} = TF/IDF)
- \mathcal{RC} : Wikipedia 2013
- $\eta_p = 0.02, \eta_i = 0.9$ (more restrictive threshold for constants)

The query process starts with the selection of the first query element which will be aligned (the semantic pivot). In the example query, ‘*Mountain*’ is the semantic pivot ($\{\text{Mountain}\} = \gamma(Q)$). Since the semantic pivot is a predicate, it is compared with the set of predicates in the G using the semantic threshold η_p over the vector space $V_{S^P\{dist\}}$:

$$\{:\text{Mountain}\} \leftarrow \bigcup_{\forall p \in P} \zeta(\text{'Mountain'}, p, \eta_p)$$

At this point $\kappa_{\mathcal{DB}} = \{:\text{Mountain}\}$.

After the alignment, the algorithm gets the next query term $\kappa_Q = \{\text{'highest'}\}$, which is an operator.

The next step is the computation of the extensional expansion of the predicate ‘Mountain’.

$$\{K2, \text{Mount Everest}, \dots\} \leftarrow \xi(: \text{Mountain})$$

Now, $\kappa_{DB} = \{ : K2, : \text{Mount Everest}, \dots \}$. The next step consists of the collection of the superset of predicates for all associated instances, to define which predicate should be aligned to the operator $\kappa_Q = \{\text{'highest'}\}$. In this step, a random sample of instances can be selected to generate a reduced superset of the predicates.

$$\{ : \text{firstAscentPerson}, : \text{locatedInArea}, : \text{elevation}, \dots \} \bigcup_{\forall i \in \{ : K2, : \text{Mount Everest}, \dots \}} \sigma_{\{ ?p(i,x) \vee ?p(x,i) \vee ?p(i) \}}$$

The semantic relatedness is computed between the query context $\kappa = \{\text{'highest'}\}$, and the set of selected predicates, defining the operator predicate alignment: $\lambda_{\text{highest}/\text{elevation}}$.

$$\{ : \text{elevation} \} \leftarrow \bigcup_{\forall p \in \pi_{\{ : \text{firstAscentPerson}, : \text{locatedInArea}, : \text{elevation}, \dots \}}} \zeta(\text{'highest'}, p, \eta_p)$$

The next step consists of the selection of the predicate alignments:

$$\{ \text{'8848 m'}, \text{'8611 m'}, \dots \} \leftarrow \sigma_{\{ : \text{Mountain}(x_0) \wedge : \text{elevation}(x_0, x_1) \}}$$

Since ‘highest’ is also a solution modifier, it has the corresponding functional component:

$$\text{highest}(\{ \text{'8848 m'}, \text{'8611 m'}, \dots \}) = \text{Top}(\{ \text{'8848 m'}, \text{'8611 m'}, \dots \}) = \text{'8848 m'}$$

Figure 8.16 and Figure 8.17 depicts the steps for answering the query over the RDF(S) graph.

8.7 The Treo System

8.7.1 Overview

The proposed schema-agnostic query model is instantiated into the *Treo* schema-agnostic system. The goal of the Treo system is to provide both a reference architecture and a prototype for the distributional semantics based schema-agnostic query approach. The Treo system was designed to be evaluated under a question answering over RDF(S) data scenario: however its core components (τDB) can be adapted to other structured data model types.

In this section, the high-level components of the Treo system are described together with the main elements of the interface of the system.

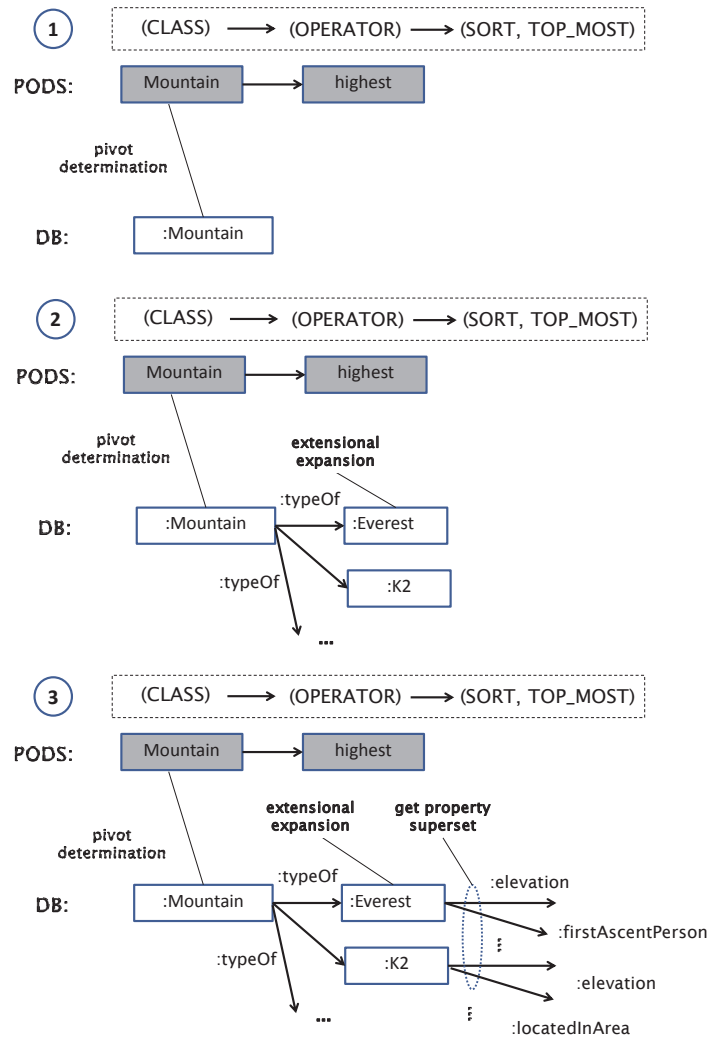


FIGURE 8.16: Execution of a query processing plan for the query ‘What is the highest mountain ?’ (Part I)

8.7.2 Architecture

The high-level workflow and main components for the query approach are depicted in Figure 8.18. The architecture is organized into three macro-components: (i) the indexer, (ii) the query analysis and (iii) and the query processing components.

The first phase consists of the *query analysis* process, which resolves a set of *PODS* and *query features* from the natural language query.

The second phase consists of the *query processing approach* which defines a sequence of search, constraints composition and solution modifier operations over the database embedded in the $\tau - Space$, based on the query plan defined by the query plan algorithm. The *Query Planner* generates the sequence of operations (the *query processing plan*) over the data graph on the semantic inverted index ($\tau - Space$). The query processing plan

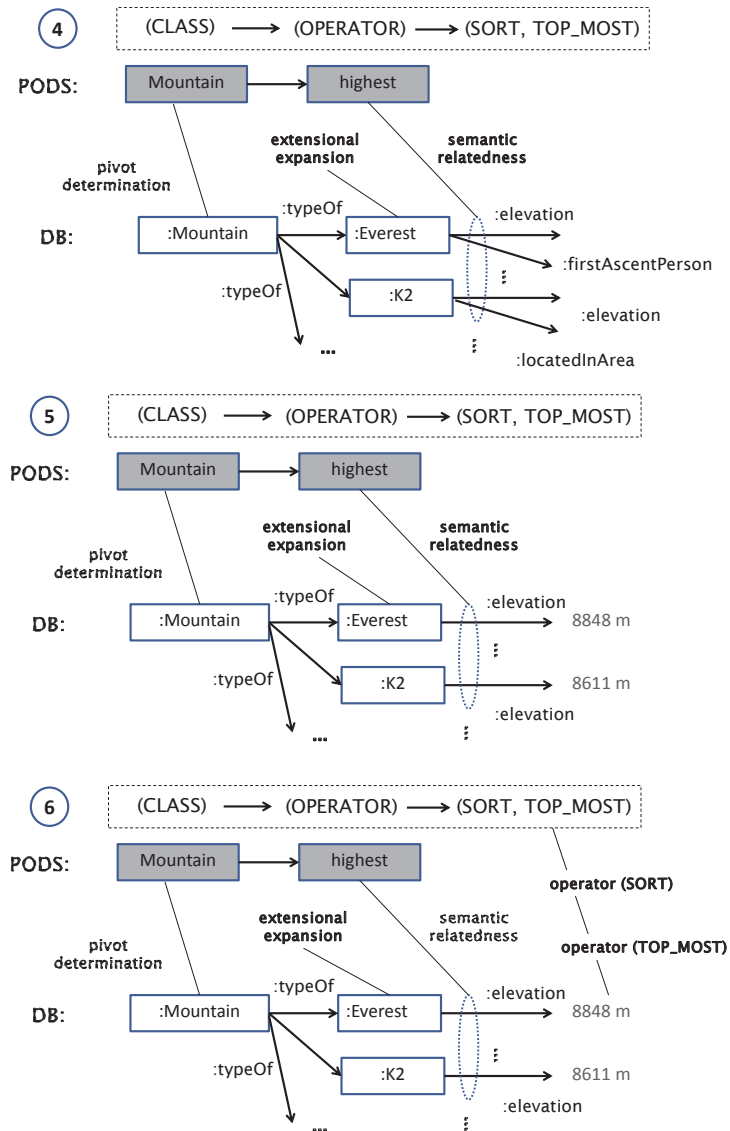


FIGURE 8.17: Execution of a query processing plan for the query 'What is the highest mountain?' (Part II)

is sent to the *Query Processor* which initially executes the *search operations* part of the query plan over the *Distributional Search* and *Instance Search* component. The query plan also includes the application of a set of *constraint composition* & *solution modifier operations* which are implemented in the *Operators* component. The result of search operations can be disambiguated using the *Disambiguation* component for *pivot entities* and *predicates*.

8.7.3 User Interface

There are three interaction modes for the query approach:

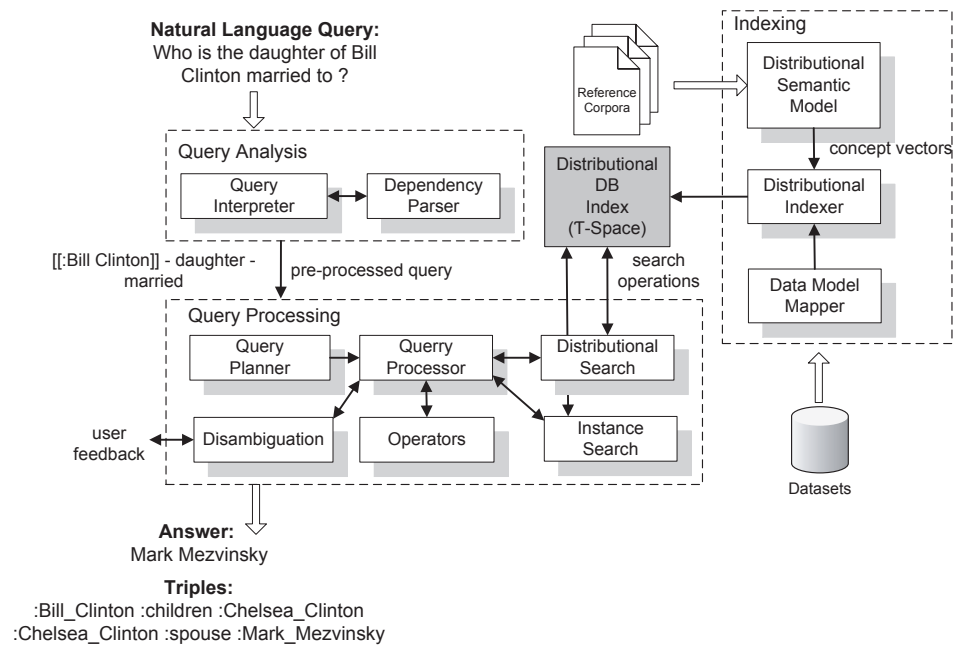


FIGURE 8.18: High-level components diagram of the schema-agnostic query approach.

- **Simple Query:** Consists of a simple query interface, where users can interact with the system by typing a natural language query. This interaction mode is targeted towards both casual users and domain experts.
- **User-Feedback Mode:** Allow users to provide feedback for the platform, disambiguating semantic pivots and predicates, and filtering out unrelated triples in the result set, allowing a simple dialog process between the user and the query processing engine. This interaction mode is targeted towards more advanced users and domain experts (e.g. data analysts).
- **Terminology Search:** Allow users to search for terminology-level elements, supporting the exploration of the types of predicates (instances and classes) in the database. This interaction mode targets data experts exploring the dataset schema.

The *simple query mode* consists of the following components:

- **Query/Search Component:** *Text Box* which allows users inputting the natural language query.
- **Search Mode Component:** *Radio button* which allows users to select between natural language query or terminology search.



FIGURE 8.19: Screenshot of the initial query interface.

- **Post-Processed Results Component:** Returns a list of post-processed answers (entities, aggregations, yes/no answers) as natural language text.
- **Data Results Component:** Returns the set of triples which is the answer or which supports the post-processed answer. This element gives the context for users to verify the suitability of the answer, supporting them in the verification of the answer. The triples are translated into a simple natural language format.
- **Picture Visualization Component:** Returns a picture associated with the answer, in case it is available in the dataset.

Figure 8.19 depicts the initial interface of the system. Figure 8.20 depicts the elements of the query interface for the example query ‘*Is Margaret Thatcher a chemist?*’, showing a post-processed result (in this example: *Yes*). Figures 8.21 and 8.22 shows the output for the two example queries. For the *query example I* (Chelsea Clinton) the semantic best-effort characteristic of the approach can be observed, where other highly related triples were returned by the distributional matching. Figures 8.23 and 8.24 show other example queries.

8.7.4 Examples

Figure 8.25 and Figure 8.26 shows the interface for the *terminology search*, displaying classes and properties for the DBpedia terminology-level elements.

The screenshot shows the Treo search engine interface. At the top left is the Treo logo. To its right is a search bar containing the query "Was Margaret Thatcher a chemist ?" and a "Search" button. Below the search bar are radio buttons for "Data" (selected) and "Vocabulary".

The search results are displayed under the heading "Was Margaret Thatcher a chemist ?". A "Short Answer" section shows a bullet point: "• yes". Below this is an "Answer" section containing a list of triples:

- Margaret Thatcher's description is Prime Minister of the United Kingdom (1979u20131990)
- Margaret Thatcher's short Description is Prime Minister of the United Kingdom
- Margaret Thatcher's type is Women Chemists
- Margaret Thatcher's subject is Category Women chemists
- Margaret Thatcher's type is English Chemists
- Margaret Thatcher's subject is Category English chemists
- Margaret Thatcher's profession is Chemist
- Margaret Thatcher's profession is Chemist

Annotations with arrows point to the search bar (Query/Search Component), the search mode options (Search Mode), the short answer (Post-Processed Results), and the list of triples (Triple Results).

FIGURE 8.20: Screenshot of the result of the Treo engine for the query 'Is Margaret Thatcher a chemist?'.
Thatcher a chemist?'.

The screenshot shows the Treo search engine interface. At the top left is the Treo logo. To its right is a search bar containing the query "Who is the daughter of Bill Clinton married t" and a "Search" button.

The search results are displayed under the heading "Who is the daughter of Bill Clinton married to ?". An "Answer" section contains a list of triples:

- Bill Clinton child Chelsea Clinton
- Bill Clinton children Chelsea Clinton
- William Jefferson Blythe, Jr. child Bill Clinton
- Virginia Clinton Kelley child Bill Clinton
- Virginia Clinton Kelley children Bill Clinton
- Chelsea Clinton parents Hillary Rodham Clinton
- Chelsea Clinton parents Bill Clinton
- Chelsea Clinton parent Bill Clinton
- Chelsea Clinton parent Hillary Rodham Clinton
- Chelsea Clinton spouse Marc Mezvinsky

To the right of the text results is a photograph of Chelsea Clinton speaking into a microphone. An annotation with an arrow points to this photograph (Picture Component).

FIGURE 8.21: Screenshot of the result of the Treo engine for the query 'Who is the daughter of Bill Clinton married to?'.
daughter of Bill Clinton married to?'.

The Treo system uses two types of *user-feedback elements*: *result filtering* and *disambiguation*. These elements are described below:

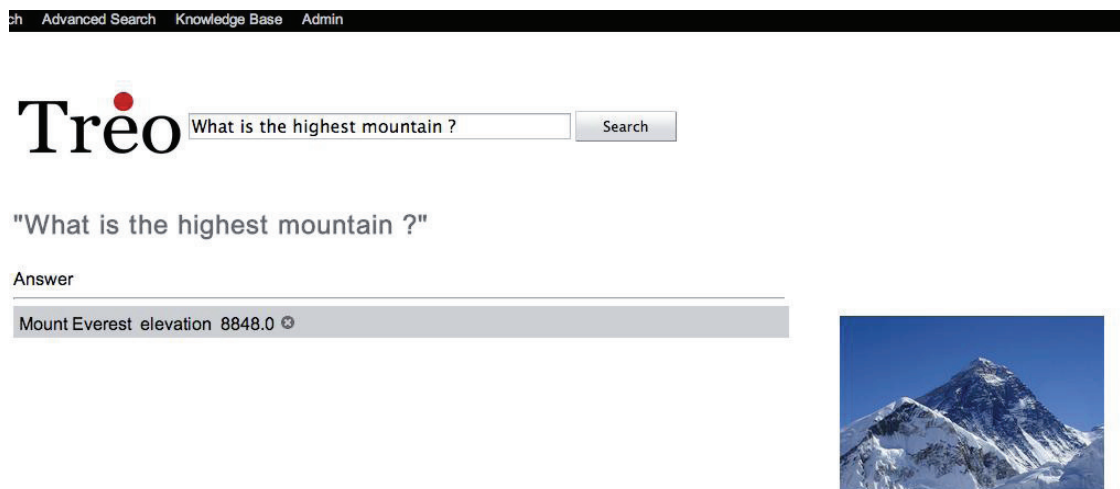


FIGURE 8.22: Screenshot of the result of the Treo engine for the query *'What is the highest mountain?'*.

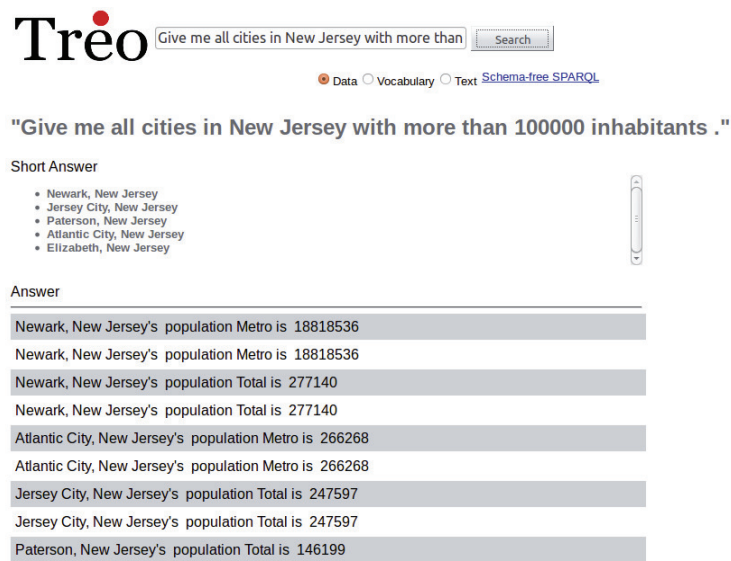


FIGURE 8.23: Screenshot of the result of the Treo engine for the query *'How tall is Claudia Schiffer?'*.

- **Result Filtering:** Supports users to remove elements from the result set (Figure 8.27).

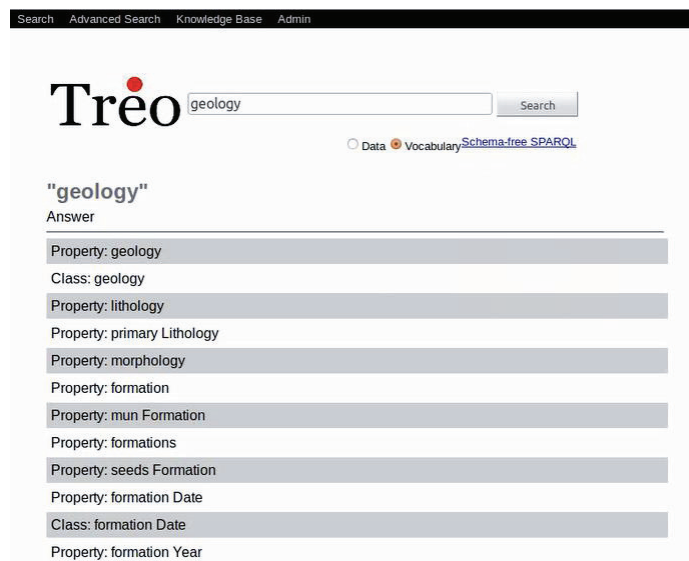
Interaction: The action is performed by a single click in one triple. The removed triple appears with a strike through its text. Users can de-select the triple by clicking on it again. Figure 8.27 shows the result of a filtering component.

- **Instance/Class (Pivot) Disambiguation:** Allow users to select/remove instance/-class semantic pivots, whenever a semantic matching is in an ambiguity range.



The screenshot shows the Treo search engine interface. At the top, the Treo logo is followed by a search bar containing the query "Give me all cities in New Jersey with more than" and a "Search" button. Below the search bar, there are radio buttons for "Data" (selected), "Vocabulary", and "Text", along with a link for "Schema-free SPARQL". The main heading is "Give me all cities in New Jersey with more than 100000 inhabitants .". Underneath, there is a "Short Answer" section with a bulleted list of five cities: Newark, New Jersey; Jersey City, New Jersey; Paterson, New Jersey; Atlantic City, New Jersey; and Elizabeth, New Jersey. To the right of this list is a vertical scrollbar. Below the "Short Answer" is an "Answer" section containing a list of ten rows, each with a city name and its population. The rows are: Newark, New Jersey's population Metro is 18818536; Newark, New Jersey's population Metro is 18818536; Newark, New Jersey's population Total is 277140; Newark, New Jersey's population Total is 277140; Atlantic City, New Jersey's population Metro is 266268; Atlantic City, New Jersey's population Metro is 266268; Jersey City, New Jersey's population Total is 247597; Jersey City, New Jersey's population Total is 247597; and Paterson, New Jersey's population Total is 146199.

FIGURE 8.24: Screenshot of the result of the Treo engine for the query ‘Give me all cities in New Jersey with more than 100000 inhabitants?’.



The screenshot shows the Treo search engine interface for a vocabulary search. At the top, there is a navigation bar with links for "Search", "Advanced Search", "Knowledge Base", and "Admin". The Treo logo is followed by a search bar containing the query "geology" and a "Search" button. Below the search bar, there are radio buttons for "Data", "Vocabulary" (selected), and "Schema-free SPARQL". The main heading is "geology". Underneath, there is an "Answer" section containing a list of ten rows, each with a property or class name. The rows are: Property: geology; Class: geology; Property: lithology; Property: primary Lithology; Property: morphology; Property: formation; Property: mun Formation; Property: formations; Property: seeds Formation; Property: formation Date; Class: formation Date; and Property: formation Year.

FIGURE 8.25: Screenshot of the vocabulary search interface for the query ‘geology’.

Interaction: The action is performed by selecting the Pivot tab at the right-most button of the screen, and by checking or unchecking instances and classes in a list containing instances or class pivots returned by the semantic approximation. The interaction is stored in the user feedback database.

- **Property Disambiguation:** Allow users to select/remove property semantic pivots, whenever a semantic matching is in an ambiguity range.

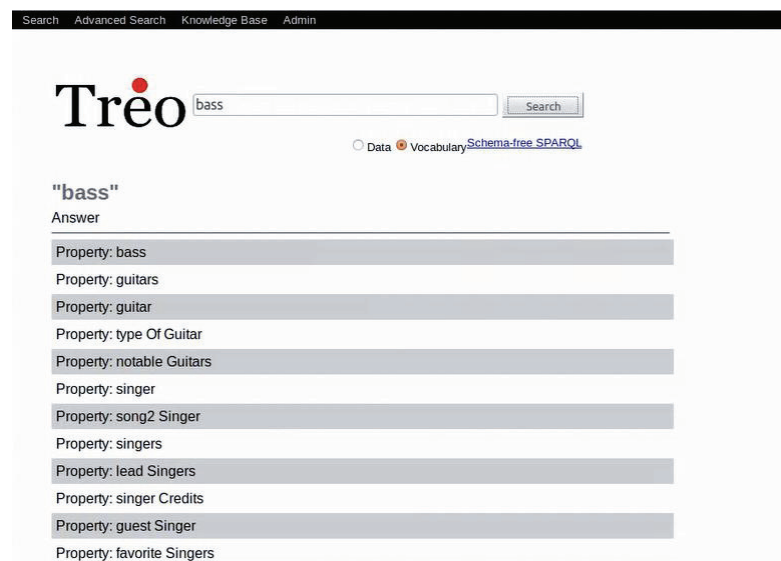


FIGURE 8.26: Screenshot of the vocabulary search interface for the query 'bass'.

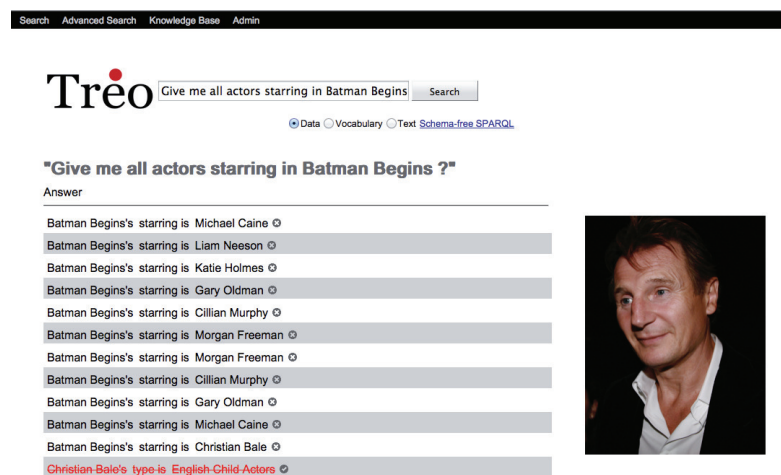


FIGURE 8.27: Result filtering component.

Interaction: The action is performed by selecting the property tab at the right-most button of the screen, and by checking or unchecking properties (using a checkbox component) in a list containing properties returned by the semantic approximation. The interaction is stored in the user feedback database.

8.7.5 Implementation

The query processing mechanism was implemented in the *Treo* schema-agnostic and NLI system following the Figure 8.18 components diagram. The system is implemented using

the Java programming language. The *semantic index* component contains the Lucene-based (Lucene 3.4¹) τ -Space implementation which can be used in other semantic search scenarios, while the *query analysis* component contains the interface for natural language queries. The web interface was developed using the Google Web Toolkit framework². Videos of the running Treo prototype can be found online³.

The *Distributional Semantic Model* used was the *Explicit Semantic Analysis* (ESA) [53] which was implemented in the *EasyESA semantic infrastructure* [219], which is described in the following section.

8.7.6 EasyESA: A Distributional Semantics Infrastructure

The construction of distributional models is dependent on the processing over large-scale corpora. The English version of Wikipedia 2014, for example, contains 44 GB of article data. The hardware and software infrastructure requirements necessary to process large-scale corpora bring high entry barriers for researchers and developers to start experimenting with distributional semantics in the context of schema-agnostic queries or in other areas. In order to facilitate the systematization on the construction and use of ESA distributional models, a distributional semantic infrastructure was built. The distributional infrastructure, named *EasyESA*, is a high-performance and easy-to-deploy distributional semantics framework and service which deploys an Explicit Semantic Analysis (ESA) [53] infrastructure.

EasyESA consists of an open source platform that can be used as a remote service or can be deployed locally. The API consists of three RESTful services:

Semantic relatedness measure: Calculates the semantic relatedness measure between two terms. The semantic relatedness measure is a real number in the $[0,1]$ interval, representing the degree of semantic proximity between two terms according to the reference corpora (Wikipedia). Semantic relatedness measures are comparative measures and are useful when sets of terms are compared in relation to their semantic proximity. Semantic relatedness can be used for semantic matching in the context of the development of semantic systems such as question answering, text entailment, event matching and semantic search.

- *Example:* Query for the semantic relatedness measure between the words *wife* and *spouse*.

¹<http://lucene.apache.org/>

²<http://www.gwtproject.org/>

³<http://bit.ly/1c36LGD>

- *Service URL:* `http://vmdeb20.deri.ie:8890/esaservice?task=esa&term1=wife&term2=spouse`
- *Result:* ‘0.0456526474’

Context vector: Given a term, it returns the associated context vector: a weighted vector of contexts (Wikipedia articles). The term can contain multiple words. The context vectors can be used to build semantic indexes (such as the τ – *Space*), which can be applied for semantic applications which depends on high performance semantic matching.

- *Example:* Query for the concept vector of the word *wife* with maximum length of 50 dimensions.
- *Service URL:* `http://vmdeb20.deri.ie:8890/esaservice?task=vector&source=wife&limit=50`
- *Result:* [“7627342, Bamboo wife, 0.2366414666”, “147083, Trophy wife, 0.2328426391”, “3516702, The Fisherman and His Wife, 0.2240851074”, “38001984, The Captain’s Wife, 0.2223930657”, “5201282, Jeremiah Mason, 0.2210460156”, “5186744, The Wife, 0.219803378”, “31516283, Second Lady of the United States, 0.2108605206”, “7200857, I Think I Love My Wife, 0.2089164555”, “21493329, My First Wife, 0.2082097977”, “473547, McMillan’s Wife, 0.2058613449”, ...]

Query explanation: Given two terms, returns the overlap between the concept vector and the ‘context windows’ for both terms on each overlapping concept. A context window for a given pair (term, concept) is a short segment from the Wikipedia article represented by the concept which contains the term.

- *Example:* Query for the concept vector overlapping between the words *wife* and *spouse*, and the context windows of both words for each concept in the overlapping dimensions.
- *Service URL:* `http://vmdeb20.deri.ie:8890/esaservice?task=explain&term1=wife&term2=spouse&limit=100` [... The position is traditionally filled by the **wife** of the president of the United States ..., ... the **wife** of the president of the United States, but, on occasion, the title ..., ... nty-fourth president; his **wife** Frances Folsom Cleveland is also counted twice ...]

EasyESA was developed using Wikirep-ESA⁴ as a basis. The software is available as an open source tool at <http://treo.deri.ie/easyesa>. The improvements targeted

⁴<https://github.com/faraday/wikirep-esa>

the following contributions: (i) major performance improvements, fundamental for the application of distributional semantics in real applications which depends on coping with high throughputs (100s of requests per second); (ii) robust concurrent queries; (iii) RESTful service API; (iv) deployment of an online service infrastructure; (v) packaging and pre-processed files for easy deployment of a local ESA infrastructure.

8.8 Chapter Summary

Using the τ – *Space* as a *semantic representation approach*, the *semantic search* and the *entropy minimization* proposed in the previous chapters, this chapter describes a *schema-agnostic query processing approach*. The query processing approach uses a set of *semantic search, composition and data transformation operations* over the τ – *Space*, which defines a schema-agnostic query processing plan. The schema-agnostic query plan defines a compositionality mechanism for resolving the query to the database facts. A supporting architecture for the query mechanism is proposed. The architecture is instantiated into the *Treo* prototype, a schema-agnostic natural language query mechanism. [198, 203, 235, 236, 237, 238, 239, 240, 241, 242, 243].

Chapter 9

Evaluation

9.1 Introduction

This chapter evaluates the schema-agnostic query approach described in the previous chapters. As already discussed in Chapter 1, the property of supporting schema-agnostic queries is implicit in the task of question answering systems over databases (QADB). QADB queries focus on scenarios which express complex information needs, where query elements need to be mapped to multiple database elements and operations. While these features are typically present in QADB evaluations, this scenario is not always present in keyword-based/semantic search over databases (KSDB), which does not always assume schema-agnosticism and more expressive/complex queries. QADB maps all the requirements for the evaluation of schema-agnostic queries over databases, with the fundamental difference that QADB systems focus on a natural language-based interaction paradigm, while schema-agnostic queries mechanisms can also explore other interaction approaches, such as *structured schema-agnostic queries*. QADB evaluation campaigns can be adapted to other types of schema-agnostic query types by mapping natural language queries to other types of schema-agnostic queries.

Due to the intersection of requirements, in this work the evaluation of the schema-agnostic query approach is grounded on existing QADB test collections. Since this work concentrates on databases with large-schema/schema-less profiles using the RDF(S) data model as a basis for discussion, the evaluation focuses on the question answering over linked data scenario (QALD), which has the support of third-party and community accepted test collections, mapping to the requirements for the evaluation of this work.

This chapter starts with the description of the evaluation methodology (Section 9.2). The suitability of the test collection to verify the main research hypotheses is analysed

in Section 9.3; Section 9.4 describes the different dimensions of the evaluation; Section 9.6 describe the results with regard to the relevance of the search results; Section 9.10 provides the comparative evaluation over existing systems; Section 9.9 provides a post-mortem analysis of the queries, analyzing the queries which were not answered, making explicit the limitations of the approach and some of the future research directions; Section 9.11 provides the coverage of the core requirements for schema-agnostic queries.

9.2 Evaluation Methodology

The evaluation methodology focuses on making explicit the steps necessary to evaluate the research hypotheses. It consists of the following steps:

1. **Test collection selection:** Selection of a candidate test collection based on the assumptions of the research hypotheses and on the core requirements.
2. **Statistical analysis:** Statistical analysis of the test collection for the verification of the suitability of the test collection for the evaluation of the research hypotheses and the core requirements coverage.
3. **Metrics selection:** Selection of evaluation metrics to quantify the suitability of the approach to the requirements.
4. **Experimental set-up and evaluation:** Development of a supporting prototype and deployment of the test collection.
5. **Schema-agnostic query approach evaluation:** Comprises the quantitative evaluation of the relevance of the results, query performance, indexing performance, index size and maintainability/transportability. The evaluation dimensions are mapped to the core requirements for a schema-agnostic query approach.
6. **Components evaluation:** Provides a quantitative and qualitative analysis of the errors associated with each query processing component of the approach.
7. **Comparative evaluation:** Compares the existing approaches to baseline QALD systems.
8. **Critical post-mortem analysis:** Provides a qualitative analysis of the queries which are either not addressed or are poorly addressed.

Additionally, the evaluation extends existing limitations in the evaluation of QADB approaches. The most prominent limitations which need to be addressed in order to measure the coverage of the set of core requirements are the following:

1. **Lack of temporal performance measurements.**
2. **Lack of dataset adaptation effort measurements (measuring the effort of manual intervention of adapting/semantically enriching the dataset at indexing time).**
3. **Lack of query interaction effort measurements (measuring the effort of manual intervention for semantically enriching/performing query disambiguation at query time).**

9.3 Test Collection Analysis

9.3.1 Motivation

The objective of this section is to ensure that the selected test collection satisfies the characteristics necessary to support the evaluation of the *thesis' hypotheses* and *core requirements*. The corroboration of the hypotheses should be supported by the evaluation methodologies and metrics and by the characteristics of the test collection in which the metrics are collected. The hypotheses are rewritten below with expressions which are dependent on the test collection properties highlighted in bold.

- **Hypothesis I:** Distributional semantics provides an accurate, comprehensive and low maintainability approach to cope with the **abstraction-level** and **lexical-level dimensions** of **semantic heterogeneity** in schema-agnostic queries over large-schema open domain datasets.
- **Hypothesis II:** The compositional semantic model defined by the query planning mechanism supports **expressive schema-agnostic queries** over **large-schema open domain datasets**.
- **Hypothesis III:** The proposed distributional-relational structured vector space model ($\tau - Space$) supports the development of a **schema-agnostic query mechanism** with interactive query execution time, low index construction time and size and scalable to **large-schema open domain datasets**.

Requirement	Verification criteria
Dataset size & semantic heterogeneity	# of classes and properties $> 10^3$ s, # of records (triples) $> 10^6$ s, # number of schema-editors $> 10^2$
Comprehensive query set	# of distinct query patterns, # of distinct query features, # of distinct mapping patterns
Query-Dataset semantic gap	even % distribution of distinct matching patterns mapping to different mapping types, # of distinct mapping patterns
Realistic & Representative query set collection	Graph patterns similar to those used in real queries , # of operators, # of vocabularies

TABLE 9.1: Requirements for the test collection and associated evaluation metrics.

These expressions are mapped into the core requirements for the test collection which describe the characteristics that should be present in the test collection to support the experimental corroboration of the hypotheses. The core test collection requirements and their mappings to the hypotheses are described below:

- *Dataset semantic size & heterogeneity*: The test collection structured dataset should reflect a large-schema/schema-less and semantically heterogeneous scenario. Maps to hypothesis context: *large-schema open domain datasets* (Hyp. I, II, III).
- *Query-Dataset semantic gap*: The test collection should manifest the semantic gap between queries and datasets, which is an intrinsic condition for evaluating schema-agnostic queries. Maps to hypotheses context: *schema-agnostic queries* (Hyp. I, II, III).
- *Comprehensive query set*: The query set should cover a representative set of query features. Maps to hypothesis context: *expressive* (Hyp. II).
- *Realistic query set collection*: Desirable condition which provides the generalization of the approach for realistic use cases (Hyp. I, II, III).

The core test collection requirements can be mapped into metrics which allow the quantitative evaluation of the test collection. This mapping is described in Table 9.1.

The main test collections for evaluating question answering and natural language interfaces over linked data/databases were pre-selected. These test collections are described below.

Data for Learning Natural Language Interfaces to Databases (Tang & Mooney, 2001) [244]:

- **Tasks**: Domain specific natural language queries in three subdomains: (i) restaurant information in N. California; (ii) job announcements posted in the newsgroup austin.jobs; (iii) a simple U.S. geography database under the Prolog and OWL data model.

- **Query set:** The U.S. geography subdomain contains 877 natural language questions. The other subdomains contain a schema of similar size.
- **Datasets:** The U.S. geography subdomain contains 9 classes, 28 properties, 697 instances. The other subdomains contain a similar number of queries.

Question Answering over Linked Data 2011 (QALD 2011) [77]:

- **Tasks:** Answer open domain and domain specific natural language queries.
- **Query set:** Training set: 50 natural language queries (DBpedia), 50 natural language queries (MusicBrainz). Testing set: 50 natural language queries (DBpedia), 50 natural language queries (MusicBrainz).
- **Datasets:** DBpedia with YAGO Links (DBpedia 3.6) [74] (Open domain), MusicBrainz (Domain specific).

Comparatively, the QALD 2011 DBpedia provides a test collection which targets a large-schema, higher heterogeneity scenario, a core criteria which was not satisfied by the Tang & Mooney test collection. The test collection is part of the Question Answering over Linked Data (QALD) challenge, a community supported challenge to provide a reproducible and comparative evaluation across different QA over Linked Data systems.

The next sections provide the quantitative analysis of the QALD test collection according to the metrics described in Table 9.1.

9.3.2 Dataset Analysis

9.3.2.1 Dataset semantic size & heterogeneity requirement

This section describes the analysis of the target dataset (DBpedia 3.6) used in the QALD 2011 test collection.

DBpedia [74] is an open domain RDF dataset derived from structured/semi-structured information present in Wikipedia, including *infoboxes* data and part of its *link structure* (e.g. category, disambiguation links). In the QALD 2011, links to YAGO categories (represented as *rdf:type* derived from the category links of Wikipedia) are included in the test collection. Since DBpedia is derived from Wikipedia, it provides an exemplar instance of a semantically heterogeneous dataset, due to the domain coverage of Wikipedia and the decentralisation of its content generation.

Metric	DBpedia 3.6
# of instances	9,434,677
# of properties	45,769
# of classes	288,316
# of triples	128,071,259
size	17GB

TABLE 9.2: Dataset metrics.

Requirement Metrics	DBpedia 3.6 Values	DBpedia 3.6 Coverage
# of classes, and properties $> 10^3$ s	$10^4 - 10^5$ s $10^4 - 10^5$	High
# of records (triples) $> 10^6$ s	10^8 s	High
# of schema editors $> 10^3$ s	10^6 s	High

TABLE 9.3: Dataset semantic size & heterogeneity requirement coverage.

DBpedia 3.6, the version which is used in the QALD 2011 test collection, has the following number of elements (Table 9.2):

Table 9.3 describes the coverage of the scale of the metrics associated with the *dataset size* & *semantic heterogeneity* requirement.

Conclusion: The DBpedia dataset provides a large-schema/schema-less and semantically heterogeneous dataset and it is appropriate for the evaluation of open domain schema-agnostic query mechanisms.

9.3.3 Query Set Analysis

9.3.3.1 Test Collection Format

The QALD 2011 query set contains 50 training and 50 test question-answer (QA) pairs. Each QA item in the QALD test collection contains three elements: (i) the natural language question ($\langle \text{string} \rangle$); (ii) the corresponding structured SPARQL query ($\langle \text{query} \rangle$); (iii) answer ($\langle \text{answers} \rangle$). Figure 9.1 depicts a *task item* of the QALD 2011 test collection. The query set and the associated SPARQL mappings are listed in Appendix A. The answers are available in ¹.

The following subsections measure different dimensions of the query set: (i) *query features analysis* and (ii) *query patterns analysis*.

¹<http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/?x=home&q=1>

```

<question id="123">
  <string>Which caves have more than 100 entrances?</string>
  <query>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX onto: <http://dbpedia.org/ontology/>
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    SELECT ?uri ?string
    WHERE {
      ?uri rdf:type onto:Cave .
      ?uri onto:numberOfEntrances ?entrance .
      FILTER (?entrance > 100) .
      OPTIONAL { ?uri rdfs:label ?string . }
      FILTER (lang(?string) = "en") }
    </query>
  <answers>
    <answer>
      <uri>http://dbpedia.org/resource/Kanheri_Caves</uri>
      <string>Kanheri Caves</string>
    </answer>
    <answer>
      <uri>http://dbpedia.org/resource/Ox_Bel_Ha_Cave_System</uri>
      <string>Ox Bel Ha Cave System</string>
    </answer>
  </answers>
</question>

```

FIGURE 9.1: Question item from the QALD 2011 test collection.

Query Features	QALD-DBpedia'2011
Contains instance reference	0.63
Contains class reference	0.12
Contains complex class reference	0.10
Contains operator reference	0.15
Contains constraint composition	0.84

TABLE 9.4: Statistics for the features of the QALD-DBpedia'2011 query set.

9.3.3.2 Structural Variability

A *comprehensive query set* should be expressed in the *structural variability* of possible user queries, mapping to different database structures and containing references to different query features and patterns. The structural variability expressed in the query set supports the evaluation of a schema-agnostic query mechanism to cope with *structural differences* between query and dataset.

Query Feature Analysis

Natural language queries over databases can be classified according to the presence of the set of features, according to references to data model types elements, database operators or different compositional patterns. In this analysis we use the set of query features which were described in Sections 8.5.5 and 8.5.4.

Table 9.4 shows the distribution of query features in the QALD 2011 query set.

Unique Query Pattern	Freq
INSTANCE PREDICATE VARIABLE	0.4489
CLASS TYPE VARIABLE VARIABLE PREDICATE VARIABLE	0.1020
VARIABLE PREDICATE VARIABLE VARIABLE TYPE COMPLEX_CLASS	0.0816
INSTANCE PREDICATE VARIABLE VARIABLE PREDICATE VARIABLE	0.0816
CLASS TYPE VARIABLE VARIABLE PREDICATE INSTANCE	0.0408
INSTANCE PREDICATE VARIABLE VARIABLE TYPE COMPLEX_CLASS DISJ	0.0408
INSTANCE PREDICATE VARIABLE VARIABLE TYPE COMPLEX_CLASS	0.0408
VARIABLE TYPE CLASS VARIABLE PREDICATE VARIABLE DISJ	0.0204
CLASS TYPE VARIABLE VARIABLE PREDICATE VARIABLE ORDER	0.0204
COMPLEX_CLASS TYPE VARIABLE VARIABLE PREDICATE VARIABLE AGGREGATE	0.0204
INSTANCE PREDICATE VARIABLE VARIABLE PREDICATE VARIABLE DISJ	0.0204
COMPLEX_CLASS TYPE VARIABLE VARIABLE PREDICATE VARIABLE ORDER	0.0204
CLASS TYPE VARIABLE VARIABLE PREDICATE INSTANCE VARIABLE PREDICATE VARIABLE DISJ	0.0204
CLASS TYPE VARIABLE VARIABLE PREDICATE INSTANCE VARIABLE PREDICATE VARIABLE ORDER	0.0204
CLASS TYPE VARIABLE VARIABLE PREDICATE INSTANCE VARIABLE PREDICATE VARIABLE	0.0204
# of unique patterns	14

TABLE 9.5: Unique query patterns (QALD 2011).

Conclusion: Each query feature category has a significant representation ($> 10\%$) in the query set. The distribution is not homogeneous and is more biased towards queries containing instances.

Query Patterns Analysis

This analysis aims at measuring the number of *unique data graph patterns* which are covered in the QALD 2011 datasets, aiming at providing an indication of the level of query expressivity expressed in the test collection. The first query categorization approach consists in the generation of query patterns based on data model types (instance, class, complex class, property, value). Table 9.5 describes the distribution of the *unique data graph patterns* for the QALD 2011 query set.

For the 50 QALD 2011 test query set there are 14 distinct query patterns. The query sets provide a comprehensive combination of different query patterns. However, the set of unique patterns are not evenly distributed. This difference in the distribution of patterns can be justified by the uneven distribution of usage patterns, where queries asking for an instance and an attribute tend to be more frequent over complex queries.

The distribution of primitive triple patterns is derived from the set of query patterns. Table 9.6 shows the set of unique references for entity type triple patterns and operator types, while Table 9.6 describes the unique triple patterns for the QALD 2011 query set.

Conclusion: The QALD 2011 query set shows a comprehensive set of query and triple patterns based on the composition of data model types and operator types.

UNIQUE TYPE PATTERNS	PROBS
INSTANCE-PREDICATE-VARIABLE	0.577
VARIABLE-PREDICATE-VARIABLE	0.207
VARIABLE-TYPE-CLASS	0.124
VARIABLE-TYPE-COMPLEX_CLASS	0.087
VARIABLE-PREDICATE-VALUE	0.004
OPERATORS	PROBS
ORDER	0.6
AGGREGATE	0.333
COMPARISON	0.067

TABLE 9.6: Unique triple patterns (QALD 2011).

The interpretation of the performance metrics should take into account the distribution of the different patterns in the dataset.

9.3.4 Conceptual/Vocabulary Gap Patterns

This section analyses the *conceptual and vocabulary gap* patterns expressed in the test collection. Two types of analysis are performed: (i) distribution of *conceptual-level* differences and (ii) distribution of *lexical categories'* differences.

Conceptual-level variation

In this analysis, the query-datasets alignments were classified according to the following categories:

- **IDENTICAL:** Query term *A* is *identical* to dataset element *B*.
- **SUBSTRING:** Query term *A* is a *substring* of the dataset element *B* or vice-versa.
- **STRING_SIMILAR:** Query term *A* has *levenshtein distance* > 0.6 in relation to dataset element *B*.
- **RELATED:** Query term *A* is *semantically related* to dataset element *B*.
- **MISSING_VOCABUALRY_MATCH:** Query term *A* does not have a corresponding dataset element *B* or vice-versa.

The analysis was performed by a *manual classification* of all *query-dataset conceptual mappings* present in the dataset. The distribution of each type of mapping is individually analyzed and categorized according to its data model type.

Conceptual/Vocabulary Gap Type	Data Model Type	%
RELATED	CLASS	0.2941
STRING_SIMILAR	CLASS	0.1176
IDENTICAL	CLASS	0.1176
SUBSTRING	CLASS	0.4705
IDENTICAL	COMPLEX_CLASS	0.5
STRING_SIMILAR	COMPLEX_CLASS	0.1
RELATED	COMPLEX_CLASS	0.4
RELATED	INSTANCE	0.0980
IDENTICAL	INSTANCE	0.6960
SUBSTRING	INSTANCE	0.1470
STRING_SIMILAR	INSTANCE	0.0490
MISSING_VOCABUALRY_MATCH	INSTANCE	0.0098
MISSING_VOCABUALRY_MATCH	NULL	1
SUBSTRING	PREDICATE	0.1680
MISSING_VOCABUALRY_MATCH	PROPERTY	0.1092
RELATED	PROPERTY	0.4117
IDENTICAL	PROPERTY	0.1680
STRING_SIMILAR	PROPERTY	0.1428
IDENTICAL	VALUE	0.25
SUBSTRING	VALUE	0.75

TABLE 9.7: Distribution of vocabulary gap types for each entity types (QALD 2011).

Table 9.7 shows the classification of *conceptual/vocabulary gap types* for different *data model types*. INSTANCES and VALUES are expected to be less bound to semantic variation. INSTANCES for example are likely to concentrate named entities, which are less bound to lexical variation or abstraction level variation, as they tend to describe specific objects. For INSTANCES, most of the vocabulary gap types should be concentrated in the (IDENTICAL, SUBSTRING or STRING SIMILARITY) categories, which is reflected in the distribution for INSTANCES. CLASSES and PROPERTIES express the description of the categories and characteristics of objects, which is likely to be shared across different objects and to be expressible under different conceptualizations. For CLASSES and PROPERTIES the semantic gap should be more evident with a large representation of the RELATED vocabulary gap category, which is confirmed in the query set analysis (Table 9.7). COMPLEX_CLASSES have more than two words making it likely that at least one of the words should vary between the query-dataset terms, which is confirmed in the query set data.

Conclusion: The query set analysis shows that the frequency distribution of *vocabulary gap types* provide a comprehensive semantic mapping scenario to evaluate different types of conceptual alignments for schema-agnostic query mechanisms. The distribution also reflects the expected vocabulary gap behavior for each data model category.

Lexical Categories

Evidence of the vocabulary gap can also be measured by analysing the differences between lexical categories from the query terms and dataset entity terms. Alignments containing different lexical categories tend to be more difficult to resolve. For this analysis task each term in the query set and the corresponding dataset terms were analysed using its lexical categories (part-of-speech (POS) tags). The POS Tags alignments were classified according to the following categories:

- **IDENTICAL:** Query POS Tag *A* is *identical* to dataset element POS Tag *B*.
- **UNMATCHED:** Query POS Tag *A* *does not match* the POS Tag of dataset element *B*.
- **PARTIAL_MATCH:** Query POS Tag *A* *partially matches* the POS Tag of element *B* and vice-versa.
- **NULL_VOCAB_ELEMENT:** There is *no corresponding matching* query-dataset explicit alignment.

Table 9.8 shows the distribution of POS Tag matching categories for each entity type. The distribution confirms a similar pattern to the previous analysis. INSTANCE and VALUE entity types tend to be expressed in the same lexical category (large incidence of alignments in the IDENTICAL and PARTIAL_MATCH). CLASS and PROPERTIES from different lexical categories are represented by the UNMATCHED class, reflecting variations in the lexical expression and abstraction-level differences.

Table 9.9 shows the total distribution of the POS Tag matching categories, including the number of distinct matchings, i.e. unique POS Tag combinations between alignment. QALD 2011 contains 57 unique POS Tag matchings (39.3% of the total matchings).

Conclusion: The QALD 2011 test collection contains a representative distribution of the variation of lexical category alignments.

9.3.4.1 Realistic & Representative Query Set: Comparative Analysis with Query Logs

The previous sections provide an analysis of the representativeness of the dataset with regard to the *distribution of query patterns* and the *distribution of query-dataset mappings*.

In order to provide additional evidence of the representativeness of the test collection, the QALD query set was compared to the USEWOD dataset [245]. USEWOD is a

POS Tag Match	Vocabulary Type	Value
PARTIAL_MATCH	CLASS	0.2941
UNMATCHED	CLASS	0.5294
IDENTICAL	CLASS	0.1764
UNMATCHED	COMPLEX_CLASS	0.3
IDENTICAL	COMPLEX_CLASS	0.5
PARTIAL _M ATCH	COMPLEX_CLASS	0.2
IDENTICAL	INSTANCE	0.7156
UNMATCHED	INSTANCE	0.1470
NULL_VOCAB_ELEMENT	INSTANCE	0.0098
PARTIAL_MATCH	INSTANCE	0.1274
NULL_VOCAB_ELEMENT	NULL	1
NULL_VOCAB_ELEMENT	PROPERTY	0.0084
UNMATCHED	PROPERTY	0.5126
NULL_QUERY_ELEMENT	PROPERTY	0.1008
PARTIAL_MATCH	PROPERTY	0.0924
IDENTICAL	PROPERTY	0.2857
IDENTICAL	VALUE	0.5
UNMATCHED	VALUE	0.5

TABLE 9.8: POS Tag matching patterns (categorized by data model types) (QALD 2011).

QALD 2011 TEST	
POS match type	prob
P(IDENTICAL)	0.5379
P(UNMATCHED)	0.2689
P(PARTIAL_MATCH)	0.1034
P(NULL_VOCAB_ELEMENT)	0.0551
P(NULL_QUERY_ELEMENT)	0.0344
Totals	
OF MATCHINGS	145
OF DISTINCT MATCHINGS	57

TABLE 9.9: POS Tag matching patterns (aggregated) (QALD 2011).

research dataset which contains query log data from different SPARQL endpoints. In this analysis the DBpedia query logs in USEWOD were analysed according to a set of features and compared to the same features in the QALD dataset. The selected set of features were: # of referenced vocabularies, # of variables, # of classes, # of instances, # of operators, # of triple patterns (per query). 19,105,182 queries from the USEWOD were analysed: 10,142,701 (DBpedia 3.8) , 7,161,159 (DBpedia 3.6) and 1,801,322 (DBpedia 3.5).

Table 9.10 contains the distribution of query features of QALD 2011 and USEWOD. Both query sets are similar in the number of instances, variables and triple patterns. QALD queries have more references to classes, while USEWOD queries contain more references to operators and use a larger number of vocabularies. A manual analysis of the DBpedia USEWOD test collection showed that the query use pattern reflected on the query logs are not fully comparable to the scenario targeted by the evaluation, where

	QALD 2011 TEST		QALD 2012 TEST		USEWOD DBPE-DIA 3.8		USEWOD DBPE-DIA 3.6		USEWOD DBPE-DIA 3.5	
measure	mean	std dev	mean	std dev	mean	std dev	mean	std dev	mean	std dev
# of vocabularies	3.26	1.05	3.08	1.18	3.91	6.62	3.2	4.68	1.96	3.96
# of variables	1.54	0.71	1.19	0.65	1.82	1.22	1.39	0.98	1.5	0.93
# of classes	0.46	0.5	0.29	0.5	0.06	0.26	0.05	0.22	0.11	0.32
# of instances	0.76	0.48	1	0.57	2.16	1.78	3.47	1.94	2.64	1.74
# of operators	0.28	0.81	0.24	0.54	2.36	4.35	1.38	4.15	2.66	4.9
# of triple patterns	1.82	1.04	1.58	0.88	1.58	1.38	1.79	0.99	1.75	1.42
# of queries	50		100		10,142,701		7,161,159		1,801,322	

TABLE 9.10: Comparative analysis between the features of QALD 2011 and USEWOD query logs.

Test Collection Requirements	QALD 2011 Coverage
Dataset size & semantic heterogeneity	High
Comprehensive query set	Medium-high
Query-Dataset semantic gap	High
Realistic & Representative query set collection	Medium-high

TABLE 9.11: Requirements for the test collection and their associated coverage.

a large percentage of the queries tend to be driven by the use of specific queries, defined inside applications.

Table 9.11 summarizes the coverage of the requirements by the QALD 2011 test collection for the DBpedia dataset.

Conclusion: The distribution of the features of the QALD test collection shares similar dimensions to USEWOD query logs. However, the USEWOD query set captures query usage patterns which can be different from the traditional question answering over linked data scenario, tending to concentrate queries which are embedded on specific applications.

9.3.4.2 Dependency of the evaluation on the QALD Dataset

The QALD dataset defines a large set of query patterns, query-dataset mappings and different topics in a large-schema setting. Additionally, the comparative analysis to the query logs shows that there are similarities between features of the QALD queries and queries issued over the DBpedia endpoint.

Requirement	Metrics
High usability & Low query construction time	query construction time
High expressivity	# of different query patterns addressed
Accurate & comprehensive semantic matching	precision, recall, f1-measure, mean reciprocal rank, # of queries answered
Low setup & maintainability effort	dataset adaptation effort (minutes), dataset specific semantic enrichment effort per query (secs), dataset specific semantic enrichment effort (minutes)
Low index size & Indexing time	index size, dataset/index size ratio, indexing time
Interactive search & Low query-execution time	query execution time, # of user interactions per query
High scalability	index construction time <i>times</i> dataset size, query execution time <i>times</i> dataset size, index size <i>times</i> dataset size

TABLE 9.12: Requirements and associated evaluation metrics.

Due to these characteristics and due to the comparability of using a community-supported test collection, QALD is used at the core of the evaluation methodology of this thesis. This defines an intrinsic dependency relation between QALD and the evaluation of this thesis. The quantitative analysis developed in the previous section aimed at making explicit the key features behind the QALD test collection, showing that QALD contains characteristics that replicate real-word schema-agnostic conditions.

Ideally a test collection for the evaluation of schema-agnostic queries would be composed of multiple datasets. This is a shortcoming of the current evaluation setting and it defines an important methodological improvement for future evaluations.

9.4 Evaluation components & performance metrics

The core goal of the evaluation is to establish the coverage of the requirements dimensions and the hypothesis for the schema-agnostic query mechanism. The analysis of the requirements coverage is mediated by the mapping of each requirement dimension to a set of evaluation metrics. Table 9.12 defines the mapping between the requirement dimensions and the evaluation metrics.

The evaluation is organized according to the categories below:

- *Relevance.*
- *Interaction & Temporal performance.*
- *Components evaluation.*

- *Comparative analysis.*
- *Critical post-mortem analysis.*

The following sections describe the experimental setup and the evaluation categories.

9.5 Evaluation Setup

The schema-agnostic approach is evaluated under an open domain question answering over Linked Data scenario, using unconstrained natural language queries. The query mechanism is instantiated in the Treo system. An independent training set of 6 questions together with 24 questions of the training set was used for the creation of the supporting prototype (Appendix D).

The query processing approach was evaluated using the Question Answering over Linked Data 2011 challenge test collection [119]. The query set contains 76 natural language queries over DBpedia 3.6 containing *rdf:type* links to YAGO classes. The dataset was indexed into the τ – *Space*. The Easy-ESA distributional infrastructure based on Wikipedia 2006 was used.

The experiments were carried on an Intel core i5-2430M CPU @ 2.40GHz computer with 8GB of RAM.

9.6 Relevance

9.6.1 Relevance Metrics

This part of the evaluation measures the relevance of the results returned by the schema-agnostic query mechanism. It consists of five metrics: *% of queries answered*, *precision*, *recall*, *f-measure*, *mean reciprocal rank* for each query and the averages for the whole query set. These measurements are described below:

% of Answered Queries measures the proportion of the queries which were answered by the query mechanism. A query is considered fully answered if $recall = 1.0$ and partially answered if $recall \geq 0.1$.

Precision provides a measure of how accurate is the answer set, i.e. the fraction of retrieved results that are relevant for the query, and it is given by the following expression:

$$p = \frac{\text{number of correct returned answers}}{\text{number of returned answers}} \quad (9.1)$$

Precision can be limited to the top-k elements returned by the query mechanism. In this case precision is defined as $p@k$ (e.g. $p@10$ precision over the top 10 elements).

The **avg. precision** is given by:

$$\text{avg. } p = \frac{1}{N} \sum_{i=1}^n p_i \quad (9.2)$$

Recall provides a measure of the completeness of the query set and consists of the fraction of relevant results that are retrieved, and it is given by the following expression:

$$r = \frac{\text{number of correct returned answers}}{\text{number of gold standard answers}} \quad (9.3)$$

The **avg. recall** is given by:

$$\text{avg. } r = \frac{1}{N} \sum_{i=1}^n r_i \quad (9.4)$$

F-measure is an harmonic mean which aggregates precision and recall into a single measure:

$$\text{f-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (9.5)$$

Reciprocal rank provides a measure of the ranking quality. The Reciprocal rank ($1/r$) of a query can be defined as the rank r at which a system returns the first relevant entity:

$$rr = \frac{1}{\text{rank number of the first relevant answer}} \quad (9.6)$$

The **Mean Reciprocal Rank** is given by:

$$\text{mrr} = \frac{1}{N} \sum_{i=1}^n rr_i \quad (9.7)$$

Measure Type	Value
Avg. Precision	0.539
Avg. Recall	0.775
Avg. F-Measure	0.561
Avg. MRR	0.431
% of queries answered	0.836
% of queries fully answered	0.627
% of queries partially answered	0.208

TABLE 9.13: Aggregate relevance results for the query results (QALD 2011 train + test).

Measure Type	Value
Avg. Precision	0.63
Avg. Recall	0.79
Avg. F-Measure	0.70
Avg. MRR	0.49
% of queries answered	0.79

TABLE 9.14: Aggregate relevance results for the query results (QALD 2011 test set).

9.6.1.1 Relevance Results Analysis

The first category of measurements evaluates the answer relevance using *mean avg. precision*, *avg. recall*, *mean reciprocal rank* (mrr) and the *% of answered queries* (fully and partially answered). The average relevance measures for the QALD queries are provided in Table 9.13 and Table 9.14. Table 9.13 contains the relevance results for QALD 2011 train + test sets which in the context of this work are used as test collections. Table 9.14 reports the relevance of the results for the QALD 2011 test collection.

80% of the queries were answered using the *schema-agnostic query processing approach*. The **0.81** recall confirms the hypothesis that the schema-agnostic query processing approach provides a comprehensive query-dataset matching mechanism. The mean avg. precision=**0.62** and mrr=**0.49** confirms the hypothesis that the approach provides an effective approximative (semantic best-effort) schema-agnostic query mechanism. The approach returns a limited list of unrelated results, where related results have higher ranking (where the correct result is between the second and the third rank position), allowing users to quickly interpret and interact with the semantically related result sets. The approach was tested under different combinations of query patterns, showing a medium-high coverage in terms of **query expressivity**.

Tables 9.15 and 9.16 shows the relevance results for each query.

Conclusion: The query approach provides a *high recall, medium-high precision* query mechanism. The high-recall provides evidence of the effectiveness of the ability to cope with the conceptual/vocabulary and structural gaps. The precision value demands a

query	p	r	f-m	mrr
Give me all actors starring in Batman Begins.	1.00	1.00	1.00	0.23
Is Christian Bale starring in Batman Begins?	1.00	1.00	1.00	1.00
Give me all soccer clubs in the Premier League.	0.05	0.85	0.10	0.20
Which countries in the European Union adopted the Euro ?	0.30	0.93	0.45	0.16
Who designed the Brooklyn Bridge ?	0.50	1.00	0.67	0.50
Which companies are located in California USA ?	0.03	0.03	0.03	0.12
Which albums contain the song Last Christmas?	1.00	0.86	0.92	0.37
When was the Battle of Gettysburg?	1.00	1.00	1.00	1.00
What is the official website of Tom Hanks?	0.00	0.00	0.00	0.00
What is the currency of the Czech Republic ?	0.50	1.00	0.67	0.50
Who was the wife of Abraham Lincoln?	0.03	1.00	0.06	0.38
Which U.S. states possess gold minerals ?	1.00	1.00	1.00	1.00
What is the profession of Frank Herbert ?	0.01	1.00	0.03	0.09
Who is called Dana?	0.40	0.89	0.56	0.04
Is Proinsulin a protein ?	1.00	1.00	1.00	1.00
In which country does the Nile start?	0.31	1.00	0.47	0.30
Which country does the Airedale Terrier come from?	1.00	1.00	1.00	1.00
Which actors were born in Germany?	0.25	0.23	0.24	0.03
Which television shows were created by Walt Disney ?	0.50	0.70	0.88	0.50
Which capitals in Europe were host cities of the summer olympic games?	0.16	1.00	0.28	0.16
What is the highest place of Karakoram ?	1.00	1.00	1.00	1.00
Which software has been published by Electronic Arts?	0.03	0.85	0.07	0.00
What did Bruce Carver die from ?	0.18	1.00	0.31	0.75
Which genre does the website DBpedia belong to?	1.00	1.00	1.00	0.52
What is the highest mountain ?	1.00	1.00	1.00	1.00
What is the area code of Berlin?	1.00	1.00	1.00	1.00
Give me the homepage of Forbes.	0.08	1.00	0.14	0.10
Who was Tom Hanks married to?	0.75	1.00	0.86	0.36
Is there a video game called Battle Chess ?	1.00	1.00	1.00	1.00
In which country is the Limerick Lake ?	1.00	1.00	1.00	1.00
Which people have as their given name Jimmy?	0.80	0.70	0.75	0.33
Who is the creator of Goofy ?	0.50	1.00	0.67	0.29
In which programming language is GIMP written?	0.43	1.00	0.60	0.61
Who created English Wikipedia ?	1.00	1.00	1.00	0.75
Through which countries does the Yenisei river flow?	1.00	1.00	1.00	0.52
Who is the owner of Aldi?	0.33	1.00	0.50	0.43
Who wrote the book The Pillars of the Earth?	0.50	0.50	0.50	0.36
Give me the capitals of all U.S. states.	0.86	0.88	0.87	0.13
Which river does the Brooklyn Bridge cross ?	0.43	1.00	0.60	0.49
Give me all films produced by Hal Roach.	0.98	0.97	0.98	0.01
Who is the author of WikiLeaks?	1.00	1.00	1.00	0.75

TABLE 9.15: Relevance results for the query results (Part I).

user interpretation effort, to filter out the incorrect answers, in a search engine-like behaviour. Additional contextual information is important to allow users to filter out incorrect results.

Impacts requirements: (i) *High usability & Low query construction time*; (ii) *High query expressivity*; (iii) *Accurate semantic matching*; (iv) *Comprehensive semantic matching*;

query	p	r	f-m	mrr
Which European countries are a constitutional monarchy ?	0.18	1.00	0.31	0.12
Who was the successor of John F. Kennedy?	1.00	1.00	1.00	1.00
What are the official languages of the Philippines ?	0.60	1.00	0.75	0.26
Who created English Wikipedia?	1.00	1.00	1.00	1.00
Which museum exhibits The Scream by Munch ?	0.50	1.00	0.67	1.00
Give me all school types.	0.93	0.93	0.93	0.04
What languages are spoken in Estonia?	0.67	1.00	0.80	0.27
Is Natalie Portman an actress?	1.00	1.00	1.00	1.00
What is the revenue of IBM?	1.00	1.00	1.00	0.75
Which are the presidents of the United States of America ?	0.98	0.95	0.97	0.10
Which books were written by Danielle Steel?	0.67	1.00	0.80	1.00
When was Lucas Arts founded?	1.00	1.00	1.00	1.00
Which people were born in Heraklion ?	0.14	0.10	0.12	0.17
Who is the owner of Universal Studios?	0.17	1.00	0.29	0.31
Which companies are in the computer software industry?	0.59	0.55	0.57	0.00
Is the wife of Barack Obama called Michelle ?	1.00	1.00	1.00	1.00
Where did Abraham Lincoln die?	0.02	1.00	0.03	0.12
Who is the mayor of New York City ?	0.20	1.00	0.33	1.00
How tall is Claudia Schiffer?	0.09	1.00	0.17	1.00
Who developed the video game World of Warcraft?	1.00	1.00	1.00	0.75
Give me all European Capitals!	0.98	0.91	0.94	0.08
Give me the birthdays of all actors of the television show Charmed.	0.00	0.00	0.00	0.00
Since when is DBpedia online?	0.50	1.00	0.67	0.50
How many films did Leonardo DiCaprio star in ?	1.00	1.00	1.00	1.00
Was U.S. president Jackson involved in a war ?	1.00	1.00	1.00	1.00
Which classis does the Millipede belong to ?	0.96	1.00	0.98	0.07
Which states border Utah?	0.00	0.00	0.00	0.00
In which films directed by Garry Marshall was Julia Roberts starring?	0.10	1.00	0.18	0.21
Which actors were born in Germany ?	0.25	0.23	0.24	0.03
Which birds are there in the United States ?	0.88	0.79	0.83	0.02
Which software has been published by Mean Hamster Software ?	1.00	1.00	1.00	0.46
When was DBpedia released ?	0.50	1.00	0.67	0.50
When was Capcom founded ?	1.00	1.00	1.00	0.61
Who is the daughter of Bill Clinton married to?	1.00	1.00	1.00	1.00
Which presidents of the United States had more than three children?	0.50	1.00	0.67	0.67
List all episodes of the first season of the HBO television series The Sopranos	0.00	0.00	0.00	0.00

TABLE 9.16: Relevance results for the query results (Part II).

9.6.2 Query Type Relevant Results

The goal of the relevance evaluation categorized by query features is to define which query features are better resolved by the query mechanism. Table 9.17 shows the categorized relevance metrics.

Queries with instances (semantic pivots) have better recall compared to *queries with classes* or *complex classes* semantic pivots. Composition operations such as path queries are addressed by the query processing mechanism. Queries containing references to operators are addressed in most of the cases. However, these results need to be interpreted in the context of the proportion of queries which have a reference to an operator.

Measure	all queries	queries with instances	queries with classes	queries with complex classes	queries with operators	path
Avg. Precision	0.61	0.65	0.77	0.46	0.88	0.63
Avg. Recall	0.87	0.95	0.76	0.67	1.00	0.88
Avg. F-Measure	0.66	0.71	0.76	0.49	0.92	0.68
Avg. MRR	0.49	0.59	0.44	0.19	0.92	0.56

TABLE 9.17: Aggregated relevance results for the query results grouped by the presence of query feature.

Conclusion: The presence of instances as a semantic pivot increases the recall of the query mechanism, in contrast with complex classes as semantic pivots. The query mechanism is able to address queries with compositional patterns such as property paths and also map to database operators. Queries targeted by the QALD 2011 test collection focuses on factoid questions.

9.6.3 Component Relevance Results

The second category of measurements in Table 9.18 evaluates individually the core search components of the approach: *instance/class(pivot) term search* and *distributional property search* for different query features. The following categories and associated metrics were evaluated:

- *Semantic Pivot Search* (avg. precision, recall, MRR)
- *Query-Vocabulary Term Matching* (avg. precision, recall, MRR)
- *Structural Matching* (accuracy)
- *User Feedback* (# of user interventions)

Queries with instances as semantic pivots have higher precision and recall compared with queries with class pivots. The individual performance of these two components minimizes the number of *user-feedback* for disambiguation over the distributional index. To support an effective user feedback dialog mechanism, the set of returned results should have high mrr and precision. From a user-interaction perspective, an average mrr higher than 0.33 (where the target result is ranked third on the list) provides a low impact disambiguation mechanism. The measured average mrr=**0.91** for *semantic pivot search* and **0.76** for *property search* components provide a low interaction cost disambiguation

Type	Measure	all queries	w/ instances	w/ classes	w/ complex classes	w/ operations	w/ const. comp.
Query Processing	Mean Avg. Precision	0.62	0.65	0.77	0.46	0.88	0.63
	Avg. Recall	0.81	0.93	0.76	0.67	1.00	0.87
	MRR	0.49	0.59	0.44	0.19	0.92	0.56
	% of queries answered	0.80	0.94	0.80	1.00	0.75	0.82
	% of queries fully answered	0.62	0.81	0.40	0.30	0.75	0.70
	% of queries partially answered	0.21	0.13	0.40	0.70	0.00	0.12
Semantic pivot Search	Avg. Entity Precision	0.47	0.49	0.56	0.27	0.36	0.49
	Avg. Semantic pivot Recall	1.00	1.00	1.00	1.00	1.00	1.00
	Semantic pivot MRR	0.91	0.96	0.73	0.82	1.00	0.90
	% of semantic pivot queries fully answered	0.88	1.00	1.00	1.00	0.75	0.88
	Avg. # of semantic pivot disambiguation operations per query	0.14	0.06	0.40	0.30	0.25	0.12
Property Search	Avg. Property Precision	0.45	0.36	0.18	0.52	0.43	0.42
	Avg. Property Recall	0.95	0.98	0.67	1.00	1.00	0.95
	Property MRR	0.76	0.81	0.30	0.40	0.71	0.83
	% of property queries fully answered	0.65	0.90	0.60	0.00	0.75	0.74
	Avg. # of property disambiguation operations per query	0.05	0.06	0.20	0.00	0.25	0.05

TABLE 9.18: Evaluation of the query processing mechanism results using natural language queries. Measures are collected for the full query mechanism and its core subcomponents: entity search and property search. The measures are categorized according to the query features.

mechanism. Both semantic pivot and property search have a high recall value (**1.0** and **0.95** respectively). Compared with queries with instances as pivots, queries containing classes as pivots have a significantly higher number of semantic pivot disambiguation operations, since classes are referenced in many different contexts and their specificity is lower. The evaluation shows that the semantic matching copes with the *ability to handle lexical variation* (including non-taxonomic and from different POS). Most queries do not require user disambiguation. The average number of user clicks per query is **0.14** for semantic pivots and **0.05** for properties. The number of entity disambiguations is higher than the number of properties' disambiguations.

Measure	value
Avg query execution time (ms)	8,530
Avg. semantic pivot search time (ms)	3,495
Avg. property search time (ms)	3,223
Avg. number of search operations per query	2.70
Avg. index insert time per triple (ms)	5.35
Avg. index size per triple (bytes)	250

TABLE 9.19: Temporal and size measures of the distributional semantic index.

9.7 Temporal & Index Size Performance Evaluation

The query approach was evaluated for its temporal performance in relation to its query and indexing time and also in relation to the size for the representation of the graph data in the distributional index. The following evaluation metrics were used:

- **Evaluation of the Query Execution Time:** avg. query execution time (ms), avg. semantic pivot search time, avg. property search time.
- **Evaluation of the Indexing Time:** avg. insert time per triple (ms).
- **Evaluation of the Indexing Size:** avg. index document size per triple (bytes).

The values of the metrics are displayed in Table 9.19.

The **8,530 ms** average query execution time supports an interactive query mechanism. Queries with the (INSTANCE - PREDICATE - VARIABLE) pattern are typically performed in less than 2,000 ms, while the longest query ‘*What is the highest mountain?*’ is performed in 53,623 ms. In this lost case, most of the query execution time concentrates in operations which are not optimized in the Treo system (e.g extensional expansion, sorting and doing a conditional filter over a set of approximately 50,000 mountain instances).

The *index construction time* is the sum of the *triple indexing time* and the *distributional vector request time*. While the avg. distributional vector request time is 82 ms per request, each distributional vector is just requested and stored in the index once per indexed term.

The distributional index size increased in 17.64 % the dataset size.

Conclusion: The query processing model and the distributional semantic index provides the basis for an interactive query mechanism. The distributional semantic vector request provides a significant overhead for the index construction time, in comparison

Measure	value
Dataset specific a priori adaptation effort (minutes)	0.00
Dataset specific semantic enrichment effort per query (secs)	0.00
Dataset specific semantic disambiguation effort per query (secs)	2.20

TABLE 9.20: Dataset adaptation effort.

with term indexing approaches. However, the distributional index construction times scales to large datasets.

Impacts requirements: (i) *Interactive search & Low query-execution time*, (ii) *High scalability*.

9.8 Transportability Evaluation

The transportability evaluation takes into account the effort involved in customizing a query mechanism to a specific dataset. Some query mechanisms require a manual customization effort (dataset adaptation) during indexing time or a semantic interaction effort during query time, which may involve tasks such as semantic enrichment and disambiguation, where the user manually defines the semantic relationship between query and dataset terms (enrichment) or selects a query-dataset alignments from a list (disambiguation). While this effort is not typically measured in the context of QA systems, the goal of this evaluation is to make this effort more explicit and comparable across different systems.

Table 9.20 describes the measures associated with the adaptation effort. The proposed approach does not require an a priori dataset adaptation effort during indexing time. Additionally, because the assumption of a high recall provided by a distributional semantics matching mechanism, the mechanism does not require a manual semantic enrichment step. The approach used disambiguation dialog for increasing mostly the precision of a small percentage of the queries.

Conclusion: The approach provides a low adaptability effort. Most of the queries did not require a disambiguation effort.

Impacts requirements: (i) *Low setup & maintainability effort*

9.9 Critical Post-mortem Analysis

In the evaluation test collection three queries had precision = 0 and recall = 0. The analysis provides an analysis and justification of the queries which are not covered by the mechanism.

Query: *Give me the birthdays of all actors of the television show Charmed.*

Analysis: ‘television show’ is resolved to a property instead of being identified as the type of the instance associated with the named entity ‘Charmed’. The error is in the query analysis phase of the pipeline (entity type identification). In case the analysis was corrected the alignments (‘actors’ → :starring) and (‘birthday’ → :birthDate) are resolved using the semantic relatedness measure with scores above the threshold, with values 0.029 and 0.017 respectively as well as the core entity-semantic pivot alignment between (‘Charmed’ → :Charmed).

Query: *List all episodes of the first season of the HBO television series The Sopranos.*

Analysis: Identifies ‘HBO television series The Sopranos’ as the core entity and does not resolve to the right semantic pivot (http://dbpedia.org/resource/The_Sopranos). The error is in the query analysis phase of the pipeline (core entity identification). The approach identifies correctly the core entity-semantic pivot alignment (‘The Sopranos’ → :The_Sopranos) and the (‘episode’ → :series) and (‘season’ → :season) alignments which have semantic relatedness values 0.165 and 1.0 respectively.

Query: *In which films directed by Garry Marshall was Julia Roberts starring?*

Analysis: The query was resolved by the query mechanism with p=0.1 and r=1.0. However, the query was not mapped to its correct structured format. The query analysis step mapped the NL query into a path query with the type pattern (INSTANCE - PROPERTY - PROPERTY - INSTANCE) instead of the star-shaped query.

Conclusion: All the major errors are concentrated in the *query analysis step*. The approach needs better identification mechanisms for entities, in particular, compositions between instances and natural language terms which define the associated class of instances (e.g. television show - Charmed). The provision of a backtracking mechanism for the selection of the semantic pivots can play a major role in the improvement of the mechanism, where different semantic pivot hypothesis are tested. Additionally, the current mechanism needs to better identify patterns which map to star-shaped queries, by improving the detection of more implicit conjunction mechanisms.

System	avg. recall	avg. precision	f-measure	% of queries answered
Treo	0.79	0.63	0.70	79%
PowerAqua	0.54	0.63	0.58	48%
FRyEa	0.48	0.52	0.50	54%
Unger et al.	0.63	0.61	0.62	-

TABLE 9.21: Comparison with existing systems for the QALD 2011 test set.

9.10 Comparative Evaluation with Existing QALD Systems

In addition to the set of metrics associated with the requirements coverage, the existing approach is compared against other state-of-the-art baseline systems.

Table 9.21 shows the comparison between the distributional approach with three baseline systems. The system outperforms the existing approaches in *recall* and *% of answered queries*, showing *equivalent precision* to the top performing system. Analyzing the query features related to the queries with f-measure < 0.1 it can be observed that most of the queries which were not answered by PowerAqua have aggregations and comparisons (53%- 9 queries) and/or reference to classes (70% - 12 queries). For Freya, the same pattern was observed: queries with aggregations and comparisons account for 50% (7 queries) of the queries with f-measure < 0.1 , while queries with reference to classes account for 64% (9 queries). Comparatively, the proposed approach is able to cope with queries containing references to aggregations and comparisons and reference to classes (accounting for 40% on the queries which were not answered - 2 queries). The results for PowerAqua and Freya can be found in Appendix E.

The difference in the results can be explained by the construction of a comprehensive query planning algorithm, which provides a mechanism to detect core query features and map them into a schema-agnostic query execution plan (which in the context of this work, defines the compositional model).

9.11 Requirements Coverage

The previous sections focused on providing a detailed evaluation of the proposed query approach, where different dimensions of the requirements were evaluated. These dimensions are summarized in Table 9.22.

Each category has the possible values: *Low*, *Medium-Low*, *Medium*, *Medium-High*, *High*.

Requirement	Coverage	Suitability of the Evaluation Setup
High usability & Low query construction time	High	Usability not explicitly covered in the evaluation (Intrinsic to open natural language interfaces)
High query expressivity	Medium-high	Medium-High
Accurate semantic matching	Medium-high	High
Comprehensive semantic matching	High	High
Low setup & maintainability effort	High	High
Interactive search & Low query-execution time	Medium-high	High
High scalability	Medium	Medium-low

TABLE 9.22: Requirements coverage of the proposed schema-agnostic query approach.

The analysis of the corroboration of the hypotheses are summarized in the *Conclusions* chapter.

9.12 Chapter Summary

This chapter describes the evaluation of the proposed schema-agnostic query approach using the *Question Answering over Linked Data (QALD 2011)* test collection. The suitability of the test collection to support the evaluation of schema-agnostic queries is verified by statistically analyzing features of the test collection related to the thesis hypotheses. The query approach is evaluated using metrics which map to the set of core requirements for schema-agnostic queries. The proposed approach, confirmed the research hypotheses and had a high coverage of the core requirements for schema-agnostic queries under a semantic best-effort scenario (*high-recall* and *medium precision*). The *post-mortem* analysis of the query mechanism shows that limitations of the approach were concentrated on the transformation of natural language queries to the query plan. The associated publication to this chapter is [236].

Chapter 10

Generalization & Further Applications

10.1 Semantic Approximations at Scale

Efficient semantic approximations and the property of schema-agnosticism can impact different areas, from *reasoning* to the construction of *schema-agnostic information systems*. In this chapter, the proposed approach is generalized into its core principles (Section 10.2.1), to facilitate the reuse of these principles in other application contexts. From the core principles, two generalizations are described: the first involving a *distributional-based semantic interpretation model* derived from the principles (Section 10.2.2) and the second is an *extension of the Semantic Web Stack* to include distributional semantics components (Section 10.6.2).

Additionally, two different application scenarios using the principles are described: first applying the proposed distributional semantics-based semantic approximation to a logic programming scenario (Section 10.4) and the second applying the proposed approach to allow approximative and selective reasoning over incomplete knowledge bases (Section 10.3). These two applications employ the same principles used in the definition of the schema-agnostic query mechanism, and can be interpreted as further outcomes of the application of schema-agnosticism and distributional-based semantic approximation to reasoning and logic programming.

10.2 Knowledge-based Semantic Interpretation

10.2.1 Core Principles

The proposed schema-agnostic query approach defines a semantic approximation process which can be summarized into the following six principles:

- **Hybrid distributional-relational semantic model:** Use of the distributional semantics as a complementary layer to the relational semantics, where the semantics of the terms in the schema is extended with the symbolic relations expressed in a reference corpora.
- **Semantic pivot:** Consists on the selection of the easiest mapping (query-data alignment) through the application of heuristic methods based on lexical categories and term specificity.
- **Context-based distributional semantic approximation:** Whenever possible, the distributional semantic approximation should be done using a semantic pivot which serves to define the semantic context and the reduction of the semantic matching space.
- **Lightweight structural assumptions on the syntactic-semantic interface:** Syntactic information is used to detect possible entities and their binary relationships. There is no a priori predicate/argument structure associated to the terms in the sentence. The predicate-argument structure is given a posteriori by the database entities. The predicate-argument structure (structural mapping) follows from the combination of the *query term-dataset entity* mapping to the partial isomorphism between the sentence syntactic relations and the dataset structure.
- **Semantic best-effort:** The approach does not assume absolute accuracy of results for an automated approximation.
- **Locality of the semantic approximation:** Most semantic approximations in the context of semantic matching for addressing the vocabulary problem have evidence likely to be expressed in corpora, targeting more local semantic relations, instead of complex chains of relations, which are more intrinsic to knowledge discovery scenarios.

These principles can be applied to other semantic approximation scenarios (outside the context of schema-agnostic queries), such as for approximate reasoning. This chapter describes two further applications of these principles in the context of *logic programming* and *approximate reasoning*.

10.2.2 Semantic Interpretation Model

The compositional-distributional model behind the semantic approximation model can be generalised into a semantic interpretation model which has the particular properties described below:

- *Coupled syntactic parsing and semantic resolution to a KB.*
- *Lexical approximation in the context of the KB.*
- *Structural approximation in the context of the KB.*

The *knowledge-based semantic interpretation* (KBSI) process starts with the syntactic parsing of the natural language sentence. The grammar should take into account the prioritization of the semantic pivot, which provides the entry-point for the interpretation of the natural language sentence. The selection of the semantic pivot is expected to maximize two elements:

- *Semantic mapping probability:* By minimizing the influence of the AVS conditions.
- *Context definition:* Selection of the sentence term which contains the most topical entity for the interpretation of the remaining sentence elements.

After the alignment of the semantic pivot, the syntactic relations define the semantic interpretation sequence. The interpretation process uses semantic approximations for the conceptual mapping, considering the structural constraint imposed by the sentence syntax. The final logical form is defined by the mapping to the KB entities and their associated predicate-argument structure.

The grammar associated with the KBSI provides mapping from lexical categories and their combinations into data model categories (e.g. instance, predicate (class, property)). Additionally, the grammar uses a set of mapping and transformation operations over the KB.

- $\zeta_{model}(t, E)$: Vector-space based semantic approximation.
- $\chi(e)$: Extensional expansion.
- $\pi(e)$: Gets the predicates associated with an instance or a class e .

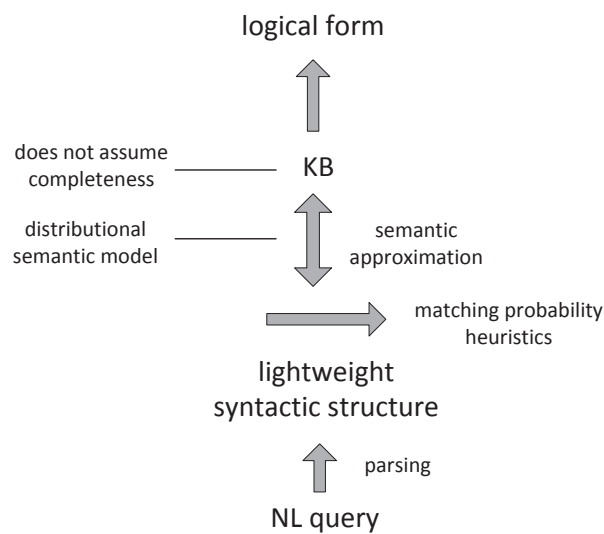


FIGURE 10.1: Schematic representation of the knowledge-based semantic interpretation model.

- $\sigma(expr)$: Variable resolution (selection).
- $op(T)$: Database functional operators {aggregation and conditional}.
- L_{op} : Logical operators {and, or}.

The application of the KB operations provides the definition a KB-based context which changes after the application of each operation. Figure 10.1 depicts a schematic representation of the KBSI model.

The KBSI model can be contrasted with traditional semantic interpretation models, which assumes a first step mapping the natural language sentence to a logical form, which is followed by a logical reasoning step using the structured knowledge base. This model requires the KB to contain the semantic representation of the sentence, overloading the structured KB with a demand for the explicit representation of the relations connecting the query terms to the database vocabulary. Figure 10.2 depicts a schematic representation of more traditional semantic interpretation models.

10.2.3 KB-based Grammar

Semantic parsing mechanisms such as *Combinatory Categorical Grammar (CCG)* can be adapted to describe the proposed semantic interpretation model. CCG defines different types of combinators, which operate in syntactically-typed lexical items [246]. The

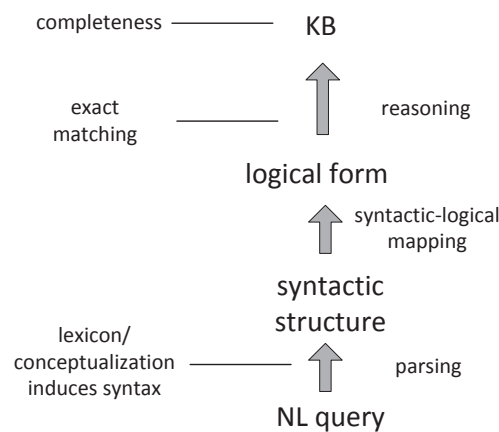


FIGURE 10.2: Schematic representation of semantic interpretation model mapping syntactic structures to logical forms.

combinators are applied in sequence until the sentence has an interpretation or proof, i.e. the derived type is the type of the whole expression.

A CCG grammar is defined by:

– Syntactic Types

- * *Primitive type*: Consists of primitive syntactic types such as S, N, or NP.
- * *Complex type*: Complex types of the form X/Y or $X \setminus Y$ are functor types which take an argument of type Y, returning an element of type X. The ‘/’ denotes that the argument should appear to the right and the ‘\’ denotes that the argument should appear on the left. Examples of complex types are: $S \setminus NP$, NP/N .

– Combinators

- * *Application*: Resolves the argument for a functor type. The application is defined as:

$$\frac{\alpha : X/Y \quad \beta : Y}{\alpha\beta : X} >$$

$$\frac{\beta : Y \quad \alpha : X/Y}{\beta\alpha : X} <$$

- * *Composition*: Compose different functor type elements. The composition is defined as:

$$\frac{\alpha : X/Y \beta : Y/Z}{\alpha\beta : X/Z} B_{>}$$

$$\frac{\beta : Y \setminus Z \alpha : X \setminus Y}{\beta\alpha : X \setminus Z} B_{<}$$

* *Type-raising*: Convert an argument type to a new functor type.

$$\frac{\alpha : X}{\alpha : T/(T)} T_{>}$$

$$\frac{\alpha : X}{\alpha : T \setminus (T/X)} T_{<}$$

In order to generate the KB-CCG (Knowledge-Based CCG), the set of data model types are included. The data model types are specific to the targeted data models. In the context of this work, the set of data model types DM is derived from RDF(S) data model types $DM = \{I, C, P, Op\}$ (instance, class, property, query operators). Different syntactic types are associated with different data model types, for example:

- NNP NNP \rightarrow I
- NN IN \rightarrow P
- ...

The mappings {syntactic type} \rightarrow {data model type} denote the possible data model correspondence of the syntactic types. One syntactic type can correspond to one or more data model types.

The set of operations O are associated with the data model types and their composition:

- I : $\zeta_I(t^I, \mathcal{I})$
- $I \setminus P$: $\zeta_{DSM}(t^P, \pi(t^I))$
- $I'P'$: $\sigma(t^I t^{P'} ? X)$
- Op : $\zeta_{DSM}(t^{Op}, \mathcal{O})$
- Op' : $op(T)$

- $C : \zeta_{DSM}(t^C, \mathcal{C})$
- $C' : \chi(e)$
- ...

After a term is resolved to a specific instance of a data model type, it is typed as a *data mapping type* using “” on the data type I' , P' , C' .

The example below shows the interpretation of a sentence under the KB-CCG.

Example I:

Query: ‘Give me the wife of Barack Obama.’, KB = DBpedia.

The CCG derivation for the example sentence is described below.

$$\begin{array}{c}
 \frac{\frac{\frac{wife}{NN} \quad \frac{of}{NN \setminus P}}{P} \quad \frac{\frac{\frac{Barack}{NNP} \quad \frac{Obama}{NNP \setminus I}}{I : \zeta_I('Barack Obama', \mathcal{I})}^G}{I'}}{P \setminus I' : \zeta_{DSM}('wife', \pi(: Barack_Obama))} < \\
 \frac{\frac{Give \ me \ the}{S/I'} \quad \frac{P' I' : \sigma(: spouse(: Barack_Obama, x))}{I'}}{S} G \\
 \hline
 S \longrightarrow >
 \end{array}$$

10.3 Further Applications: Approximate and Selective Commonsense Reasoning

10.3.1 Introduction

With the evolution of open data, better information extraction frameworks and crowd-sourcing tools, large-scale structured KBs are becoming more available. This data can be used to provide commonsense knowledge for semantic applications. However, reasoning over this data demands approaches which are able to cope with large-scale, semantically heterogeneous and incomplete KBs. In this section the principles behind the schema-agnostic approach are applied to support selective and approximative commonsense reasoning over large-scale commonsense KBs.

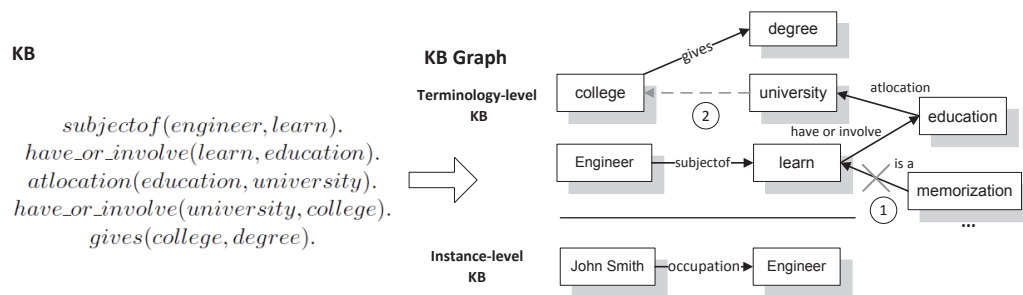


FIGURE 10.3: (1) Selection of meaningful paths, (2) Coping with information incompleteness.

10.3.2 Motivational Scenario

As a motivational scenario, suppose we have a KB with the following fact: ‘*John Smith is an engineer*’ and suppose the query ‘*Does John Smith have a degree?*’ is issued over the KB. A complete KB would have the rule ‘*Every engineer has a degree*’, which would materialize ‘*John Smith has a degree*’. For large-scale and open domain commonsense reasoning scenarios, model completeness and full materialization cannot be assumed. In this case the information can be embedded in other facts in the KB (Figure 10.3). The example sequence of relations between *engineer* and *degree* defines a path in a large-scale graph of relations between predicates, which is depicted in Figure 10.3.

In a large-scale KB, full reasoning can become unfeasible. A commonsense KB would contain vast amounts of facts and a complete inference over the entire KB would not scale to its size. Furthermore, while the example path is a meaningful sequence of associations for answering the example query, there is a large number of paths which are not meaningful under a specific query context. In Figure 10.3(1), for example, the reasoning path which goes through (1) is not related to the goal of the query (the relation between *engineer* and *degree*) and should be eliminated. Ideally, a query and reasoning mechanism should be able to filter out facts and rules which are unrelated to the reasoning context. The ability to select the minimum set of facts which should be applied in order to answer a specific user information need is a fundamental element for enabling reasoning capabilities for large-scale commonsense knowledge bases.

Additionally, since information completeness of the KBs cannot be guaranteed, one missing fact in the KB would be sufficient to block the reasoning process. In Figure 10.3(2) the lack of a fact connecting *university* and *college* eliminates the possibility of answering the query. Ideally, reasoning mechanisms should be able to cope with some level of KB incompleteness, approximating and filling the gaps in the KBs.

This application scenario describes a *selective reasoning approach* which uses a *hybrid distributional-relational semantic model* to address the problems previously described. In the scenario, DSMs are used as a complementary semantic layer to the relational model, which supports semantic approximation and coping with incompleteness.

10.3.3 Embedding the Commonsense KB into the τ -Space

We consider that a commonsense knowledge base KB is formed by a set of *concepts* $\{v_1, \dots, v_n\}$ and a set of *relations* $\{r_1, \dots, r_m\}$ between these concepts, both represented as words or short phrases in natural language. Formally, a commonsense knowledge base KB is defined by a *labeled digraph* $G_{KB}^{label} = (V, R, E)$, where $V = \{v_1, \dots, v_n\}$ is a set of nodes, $R = \{r_1, \dots, r_m\}$ is a set of relations and E is a set of directed edges (v_i, v_j) labeled with relation $r \in R$ and denoted by (v_i, r, v_j) . Alternatively, we can simplify the representation of the KB ignoring their relation labels.

Definition 10.1. Let KB be a commonsense knowledge base and $G_{KB}^{label} = (V, R, E)$ be its labeled digraph representation. A simplified representation of KB is defined by a *digraph* $G_{KB} = (V', E')$, where $V' = V$ and $E' = \{(v_i, v_j) : (v_i, r, v_j) \in E\}$.

Given the (labeled) graph representation of KB , we have to embed it into the τ -Space. To do that we have to translate the nodes and edges of the graph representation of KB into a vector representation in VS^{dist} . The vector representation of $G_{KB}^{label} = (V, R, E)$ in VS^{dist} is $\vec{G}_{KB^{dist}}^{label} = (\vec{V}_{dist}, \vec{R}_{dist}, \vec{E}_{dist})$ such that:

$$\vec{V}_{dist} = \{\vec{v} : \vec{v} = \sum_{i=1}^t u_i^v \vec{c}_i, \text{ for each } v \in V\} \quad (10.1)$$

$$\vec{R}_{dist} = \{\vec{r} : \vec{r} = \sum_{i=1}^t u_i^r \vec{c}_i, \text{ for each } r \in R\} \quad (10.2)$$

$$\vec{E}_{dist} = \{(\vec{r} - \vec{v}_i, \vec{v}_j - \vec{r}) : \text{for each } (v_i, r, v_j) \in E\} \quad (10.3)$$

u_i^v and u_i^r are defined by the weighting scheme over the distributional model¹.

10.3.4 Distributional Navigation Algorithm

Once the KB is embedded into the τ -Space, the next step is to define the navigational process in this space that corresponds to a selective reasoning process in the KB . The

¹Reflecting the word co-occurrence pattern in the reference corpus

navigational process is based on the semantic relatedness function defined as: $sr : VS^{dist} \times VS^{dist} \rightarrow [0, 1]$ is defined as:

$$sr(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2) = \cos(\theta) = \vec{\mathbf{p}}_1 \cdot \vec{\mathbf{p}}_2$$

A threshold $\eta \in [0, 1]$ can be used to establish the desired semantic relatedness between two vectors: $sr(\vec{\mathbf{p}}_1, \vec{\mathbf{p}}_2) > \eta$.

The information provided by the semantic relatedness function sr is used to identify elements in the KB with a similar meaning from the reference corpus perspective. The threshold is calculated following the semantic differential approach. Multiword phrases are handled by calculating the centroid between the concept vectors defined by each word.

Algorithm 12 is the Distributional Navigation Algorithm (DNA) which is used to find, given two semantically related terms *source* and *target* with respect to a threshold η , all paths from *source* to *target*, with length l , formed by concepts semantically related to *target* with respect to η .

The *source* term is the first element in all paths (*line 1*). From the set of paths to be explored (*ExplorePaths*), the DNA selects a path (*line 5*) and expands it with all neighbors of the last term in the selected path that are semantically related wrt threshold η and that does not appear in that path (*line 7-8*). The stop condition is $sr(\mathbf{target}, \mathbf{target}) = 1$ (*line 10-11*) or when the maximum path length is reached.

The paths $p = \langle t_0, t_1, \dots, t_l \rangle$ (where $t_0 = \mathbf{source}$ and $t_l = \mathbf{target}$) found by DNA are ranked (*line 14*) according to the following formula:

$$rank(p) = \sum_{i=0}^l sr(\vec{\mathbf{t}}_i, \vec{\mathbf{target}}) \quad (10.4)$$

Algorithm 13 can be modified to use a heuristic that allows to expand only the paths for which the semantic relatedness between all the nodes in the path and the target term increases along the path. The differential in the semantic relatedness for two consecutive iterations is defined as $\Delta_{target}(t_1, t_2) = sr(\vec{\mathbf{t}}_2, \vec{\mathbf{target}}) - sr(\vec{\mathbf{t}}_1, \vec{\mathbf{target}})$, for terms t_1, t_2 and *target*. This heuristic is implemented by including an extra test in the line 7 condition, i.e., $\Delta_{target}(t_k, n) > 0$.

Algorithm 12 Distributional Navigation Algorithm**INPUT**

- *threshold*: η
- *pair of terms* (*source*, *target*) such that $sr(\overrightarrow{\text{source}}, \overrightarrow{\text{target}}) > \eta$
- *path length*: l

OUTPUT

RankedPaths: a set of ranked score paths $\langle (t_0, \dots, t_l), \text{score} \rangle$ such that $t_0 = \text{source}$ and $t_l = \text{target}$

```

1:  $t_0 \leftarrow \text{source}$ 
2:  $\text{Paths} \leftarrow \emptyset$ 
3:  $\text{ExplorePaths} \leftarrow [(\langle t_0 \rangle, sr(\overrightarrow{t_0}, \overrightarrow{\text{target}}))]$ 
4: while  $\text{ExplorePaths} \neq \emptyset$  do
5:   remove  $(\langle t_0, \dots, t_k \rangle, sr(\overrightarrow{t_k}, \overrightarrow{\text{target}}))$  from  $\text{ExploredPaths}$ 
6:   if  $k < l - 1$  then
7:     for all  $(n \in \text{neighbors}(t_k) : sr(\overrightarrow{n}, \overrightarrow{\text{target}}) > \eta \text{ and } n \notin \{t_0, \dots, t_k\})$  do
8:       append  $(\langle t_0, \dots, t_k, n \rangle, sr(\overrightarrow{n}, \overrightarrow{\text{target}}))$  to  $\text{ExplorePaths}$ 
9:     end for
10:  else if  $k = l - 1$  then
11:    append  $(\langle t_0, \dots, t_k, \text{target} \rangle, 1)$  to  $\text{Paths}$ 
12:  end if
13: end while
14:  $\text{RankedPaths} \leftarrow \text{sort}(\text{Paths})$ 
15: return  $\text{RankedPaths}$ 

```

10.3.5 Evaluation

10.3.5.1 Setup

In order to evaluate the proposed approach, the τ -Space was built using the *Explicit Semantic Analysis* (ESA) as the distributional model.

ConceptNet[247] was selected as the commonsense knowledge base. *ConceptNet* is a semantic network represented as a labeled digraph $G_{\text{ConceptNet}}^{\text{label}}$ formed by a set of nodes representing concepts and a set of labeled edges representing relations between concepts. ConceptNet is built by using a combination of approaches, including open information extraction tools, crowd-sourced user input and open structured data. Concepts and relations are presented in the form of words or short natural language phrases. The bulk of the semantic network represents relations between predicate-level words or expressions. Different word senses are not differentiated. Two types of relations can be found: (i) recurrent relations based on a lightweight ontology used by ConceptNet (e.g. *partOf*) and (ii) natural language expressions entered by users and open information extraction tools. These characteristics make ConceptNet a heterogeneous commonsense knowledge base. For the experiment, all concepts and relations that were not in English terms were removed. The total number of triples used on the evaluation was 4,797,719. The distribution of the number of clauses per relation type is as follows: = 1 (**45,311**), $1 < x < 10$ (**11,804**), $10 \leq x < 20$ (**906**), $20 \leq x < 500$ (**790**), ≥ 500 (**50**).

TABLE 10.1: # of clauses per relation frequency.

Number of Triples	Number of Relations
= 1	45.311
$1 < x < 10$	11.804
$10 \leq x < 20$	906
$20 \leq x < 500$	790
≥ 500	50

TABLE 10.2: Top-12 frequent relations in the ConceptNet

Relation	Number of Triples
instanceof	918.123
isa	201.710
hasproperty	120.961
subjectof	96.566
definedas	94.775
relatedto	88.922
directobjectof	87.946
usedfor	62.242
have_or_involve	49.967
atlocation	49.216
derivedfrom	40.403
capableof	38.811
synonym	34.974
hassubevent	27.366
hasprerequisite	25.160
causes	18.688
motivatedbygoal	16.178
be_in	15.143
be_near	12.744
be_not	11.777
receivesaction	11.095
hasa	10.048
partof	7.104

A test collection consisting of 45 (*source*, *target*) word pairs were manually selected using pairs of words which are semantically related under the context of the Question

Answering over Linked Data challenge (QALD 2011/2012)². Each pair establishes a correspondence between question terms and dataset terms (e.g. ‘What is the *highest* mountain?’ where *highest* maps to the *elevation* predicate in the dataset). 51 pairs were generated in total.

For each word pair (a, b) , the navigational algorithm 13 was used to find all paths with lengths 2, 3 and 4 above a fix threshold $\eta = 0.05$, taking a as source and b as target and vice-versa, accounting for a total of 102 word pairs. All experimental data is available online³.

10.3.5.2 Reasoning Selectivity

The first set of experiments focuses on the measurement of the selectivity of the approach, i.e. the ability to select paths which are related and meaningful to the reasoning context. Table 10.3 shows the average *selectivity*, which is defined as the ratio between the *number of paths selected using the reasoning algorithm 13* by the *total number of paths* for each path length. The total number of paths was determined by running a depth-first search (DFS) algorithm.

For the size of ConceptNet, paths with length 2 return an average of 5 paths per word pair. For this distance most of the returned paths tend to be strongly related to the word pairs and the selectivity ratio tend to be naturally lower. For paths with length 3 and 4 the algorithm showed a very high selectivity ratio (0.153 and 0.0192 respectively). The exponential decrease in the selectivity ratio shows the scalability of the algorithm with regard to selectivity. Table 10.3 shows the average selectivity for DNA. The variation of DNA with the Δ criteria, compared to DNA, provides a further selectivity improvement ($\phi = (\# \text{ of spurious paths returned by DNA} / \# \text{ of spurious paths returned by DNA} + \Delta)$) $\phi(\text{length}2) = 1$, $\phi(\text{length}3) = 0.49$, $\phi(\text{length}4) = 0.20$.

TABLE 10.3: Selectivity

Path Length	Average Selectivity Algorithm 1	% Pairs of Words Resolved	Path Accuracy
2	0,602	0,618	0,958
3	0,153	0,726	0,828
4	0,019	0,794	0,736

²<http://www.sc.cit-ec.uni-bielefeld.de/qald-1>

³<http://bit.ly/1p3PmHr>

10.3.5.3 Semantic Relevance

The second set of experiments focuses on the determination of the *semantic relevance of the returned nodes*, which measures the expected property of the distributional semantic relatedness measure to serve as a heuristic measure for the selection of meaningful paths.

A gold standard was generated by two human annotators which determined the set of paths which are *meaningful* for the pairs of words using the following criteria: (i) all entities in the path are highly semantically related to both the source and target nodes and (ii) the entities are not very specific (unnecessary presence of instances, e.g. *new york*) or very generic (e.g. *place*) for a word-pair context. Only senses related to both source and target are considered meaningful.

The accuracy of the algorithm for different path lengths can be found in Table 10.3. The *high accuracy* reflects the effectiveness of the distributional semantic relatedness measure in the selection of meaningful paths. A systematic analysis of the returned paths shows that the decrease in the accuracy with the increase on path size can be explained by the higher probability on the inclusion of instances and classes with high abstraction levels in the paths.

From the paths classified as not related, 47% contained entities which are too specific, 15.5% too generic and 49.5% were unrelated under the specific reasoning context. This analysis provides the directions for future improvements of the approach (inclusion of filters based on specificity levels).

10.3.5.4 Addressing Information Incompleteness

This experiment measures the suitability of the distributional semantic relatedness measure to cope with KB incompleteness (gaps in the KB). 39 $\langle source, target \rangle$ entities which had paths with length 2 were selected from the original test collection. These pairs were submitted as queries over the ConceptNet KB indexed on the VS^{dist} and were ranked by the semantic relatedness measure. This process is different from the distributional navigational algorithm, which uses the relation constraint in the selection of the neighbouring entities. The distributional semantic search mechanism is equivalent to the computation of the semantic relatedness between the query (*source target*) and all entities (nodes) in the KB. The threshold criteria take the top 36 elements returned.

Two measures were collected. *Incompleteness precision* measures the quality of the entities returned by the semantic search over the KB and it is given by $incompleteness\ precision = \# \text{ of strongly related entities} / \# \text{ of retrieved entities}$. The determination

of the *strongly related entities* was done using the same methodology described in the classification of the semantic relevance. In the evaluation, results which were not highly semantically related to both source and target and were too specific or too generic were considered incorrect results. The **avg. incompleteness precision value of 0.568** shows that the ESA-based distributional semantic search provides a feasible mechanism to cope with KB incompleteness, suggesting the discovery of highly related entities in the KB in the reasoning context. There is space for improvement by the specialization of the distributional model to support better word sense disambiguation and compositionality mechanisms.

The *incompleteness coefficient* provides an estimation of the incompleteness of the KB addressed by the distributional semantics approach and it is determined by *incompleteness coefficient = # of retrieved ConceptNet entities with an explicit association / # of strongly related retrieved entities*. The **average incompleteness value of 0.039** gives an indication of the level of incompleteness that commonsense KBs can have. The *avg. # of strongly related entities* returned per query is 19.21.

TABLE 10.4: Incompleteness level.

Avg. Incompleteness. Precision	Avg. Incompleteness. Coefficient
0.568	0.039

An example of the set of new entities suggested by the distributional semantic relatedness for the pair $\langle \textit{mayor}, \textit{city} \rangle$ are: **council, municipality, downtown, ward, incumbent, borough, reelected, metropolitan, city, elect, candidate, politician, democratic.**

The evaluation shows that distributional semantics can provide a principled mechanism to cope with KB incompleteness, returning highly related KB entities (and associated facts) which can be used in the reasoning process. The level of incompleteness of an example commonsense KB was analyzed and found to be high, confirming the relevance of this problem under the context of reasoning over commonsense KBs.

10.3.6 Analysis of the Algorithm Behavior

Figure 10.4 contains a subset of the paths returned from an execution of the algorithm for the word pair $\langle \textit{battle}, \textit{war} \rangle$ merged into a graph. Intermediate nodes (words) and edges (higher level relations) provide a meaningful connection between the source and target nodes. Each path has an associated score which is the average of the semantic relatedness measures, which can serve as a ranking function to prioritize paths which are potentially more meaningful for a reasoning context. The output paths can be

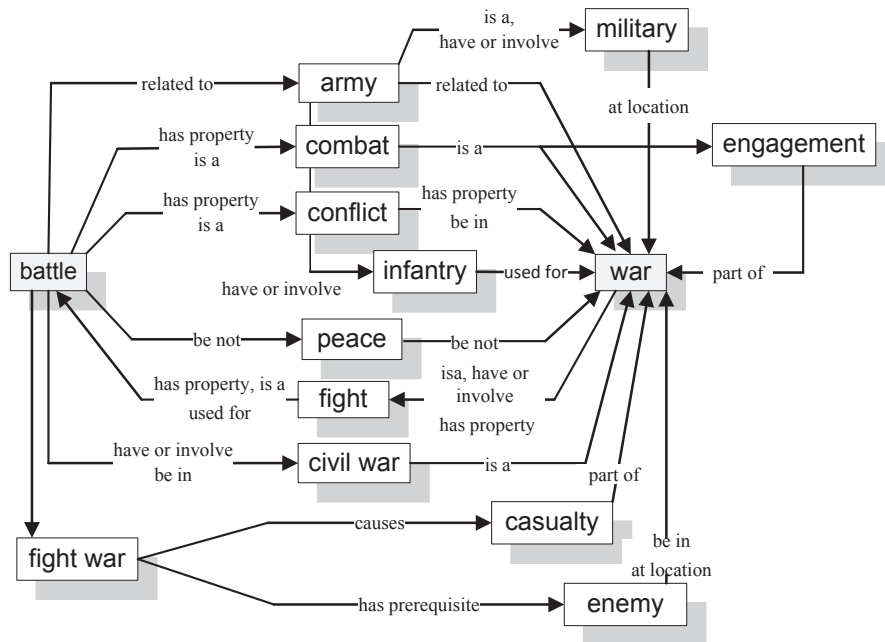


FIGURE 10.4: Contextual (selected) paths between battle and war.

interpreted as an *abductive* process between the two words, providing a semantic justification under the structure of the relational graph. Table 10.7 shows examples of paths for lengths 2, 3 and 4. Nodes are connected through relations which were omitted.

TABLE 10.5: Examples of semantically related paths returned by the algorithm (Part I).

Paths - Length 2
daughter, parent, child
episode, show, series
country, continent, europe
mayor, politician, leader
video_game, computer_game, software
long, measure, length
husband, married_man, spouse
artist, draw, paint
city, capital, country
jew, temple, religion

The selectivity provided by the use of the distributional semantic relatedness measure as a node selection mechanism can be visualized in Figure 10.5, where the distribution of the # of occurrences of the semantic relatedness values (y-axis) are shown in a logarithmic scale. The semantic relatedness values were collected during the navigation process for all comparisons performed during the execution of the experiment. The graph shows

TABLE 10.6: Examples of semantically related paths returned by the algorithm (Part II).

Paths - Length 3
club , team, play, football
chancellor , politician, parliament, government
spouse , family, wed, married
actress , act_in_play , go_on_stage, actor
film , cinema, watch_movie, movie
spouse , wife, marriage, husband
aircraft , fly, airplane, pilot
country , capital, national_city, city
chancellor , head_of_state, prime_minister, government

TABLE 10.7: Examples of semantically related paths returned by the algorithm.

Paths - Length 4
music , song, single, record, album
episode , show, series
chancellor , politician, parliament, government
soccer , football, ball, major_league, league
author , write, story, fiction, book
artist , create_art, work_of_art, art, paint
place , locality, localize, locate, location
jew , religion, ethnic_group, ethnic, ethnicity
war , gun, rifle, firearm, weapon
pilot , fly, airplane, plane, aircraft
chancellor , member, cabinet, prime_minister, government

the discriminative efficiency of semantic relatedness, where just a tiny fraction of the entities in paths of length 2, 3, 4 are selected as semantically related to the target.

In Figure 10.6 the average increase on the semantic relatedness value as the navigation algorithm approaches the target is another pattern which can be observed. This smooth increase can be interpreted as an indicator of a meaningful path, where semantic relatedness value can serve as a heuristic to indicate a meaningful approximation from the target word. This is aligned with the increased selectivity of the Δ (semantic relatedness differential) criteria.

In the DNA algorithm, the semantic relatedness was used as a heuristic in a greedy search. The worst-case time complexity of a DFS is $O(b^l)$, where b is the branching factor and l is the depth limit. In this kind of search, the amount of performance improvement depends on the quality of the heuristic. In Table 10.3 we showed that as the depth limit increases, the selectivity of DNA ensures that the number of paths does not increase by the same amount. This indicates that the distributional semantic relatedness can be an effective heuristic when applied to the selection meaningful paths to be used in a reasoning process.

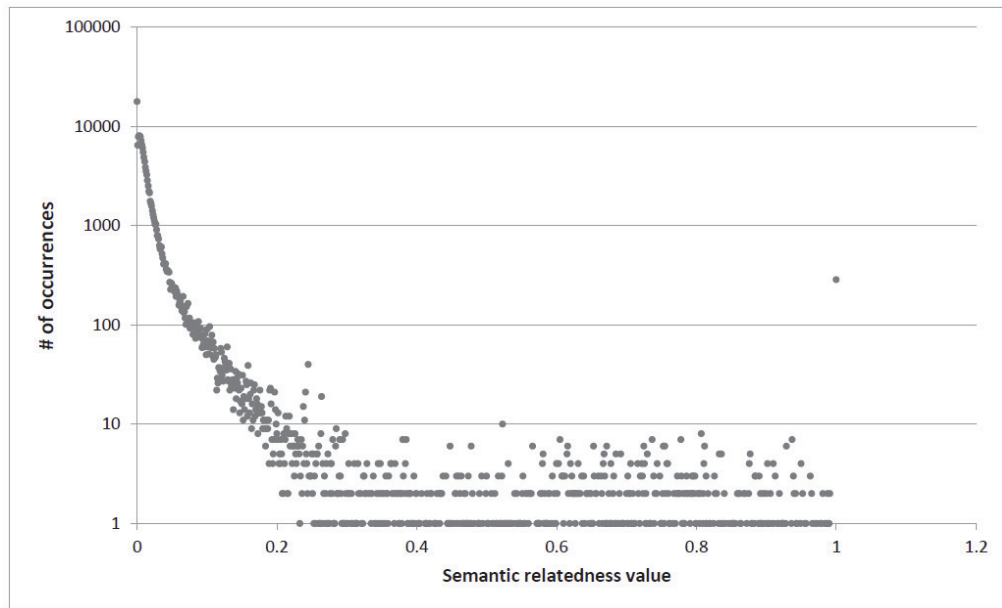


FIGURE 10.5: # of occurrences for pairwise semantic relatedness values, computed by the navigational algorithm for the test collection (paths of length 2, 3, 4). Semantic relatedness values for nodes from distances 1, 2, 3 from the source: increasing semantic relatedness to the target.

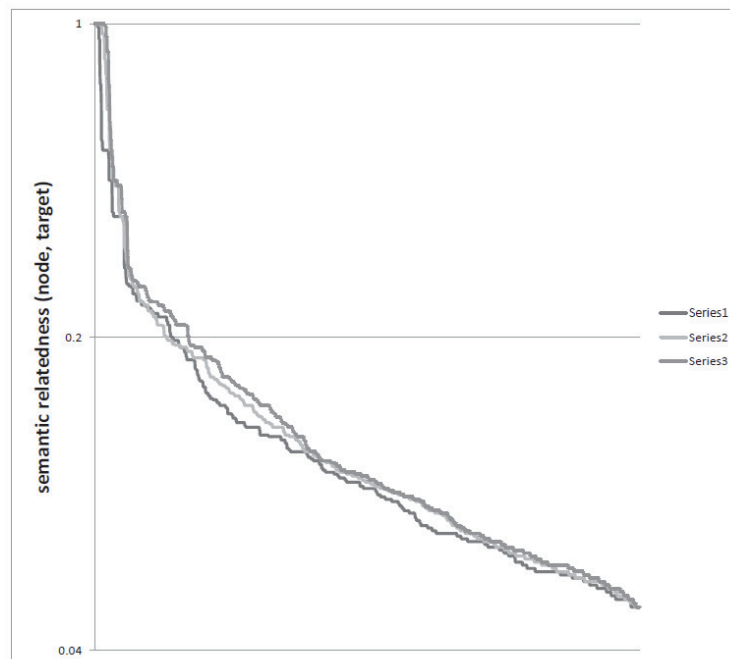


FIGURE 10.6: Increasing variation of the semantic relatedness values as navigated nodes approach the target node.

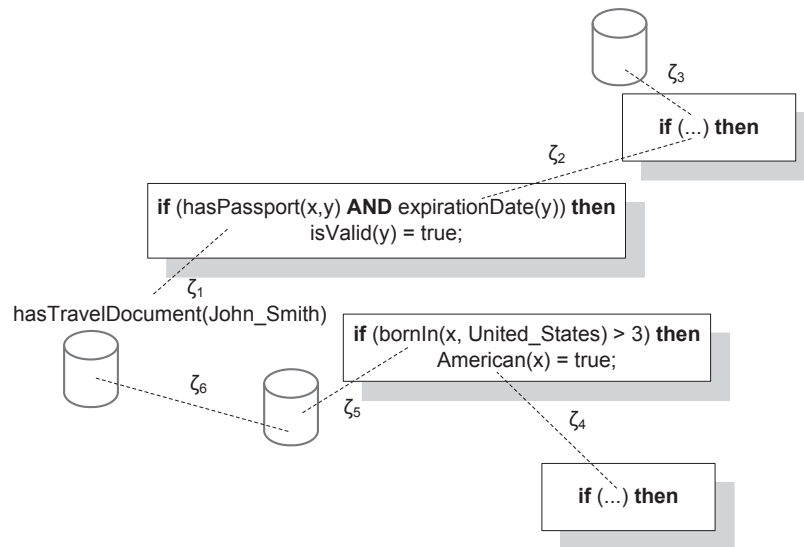


FIGURE 10.7: Depiction of a set of distributional program-database alignments.

10.4 Further Applications: Distributional Logic Programming

10.4.1 Motivation

The open communication scenario (Section 4.2.2) can be generalised from the database querying scenario to the programming scenario, where the schema-agnostic abstraction layer can be introduced between database and programs or between programs created under different contexts. The ability to define distributional semantics-based approximation as a first-class citizen in software programs provides a natural generalization of schema-agnosticism (Figure 10.7), which may facilitate the reuse and integration of software systems.

This section provides a first-level exploration on schema-agnosticism applied to programs, extending the discussion from facts and queries (databases) to rules. This discussion is done in the context of logic programming.

10.4.2 Motivational Scenario

Every knowledge or information artifact (from unstructured text to structured knowledge bases) maps to an implicit or explicit set of user intents and semantic context patterns. The multiplicity of contexts where open domain and commonsense knowledge bases can be used, defines the intrinsic semantic heterogeneity for these scenarios.

Different levels of conceptual abstraction or lexical expressions in the representation of predicates and constants are examples where a semantic/terminological gap can strongly impact the inference process.

In the scenario below an user executes a *schema-agnostic query* over a logic program Π .

Consider the query ‘*Is the father in law of Bill Clinton’s daughter a politician?*’ that can be represented as the logical query:

$$? - \text{daughter_of}(X, \text{bill_clinton}), \text{politician}(Y), \text{father_in_law}(Y, X)$$

Let us assume that the logic program Π contains facts and rules such as:

$$\begin{aligned} & \text{child_of}(\text{chelsea_clinton}, \text{bill_clinton}). \\ & \text{child_of}(\text{marc_mezvinsky}, \text{edward_mezvinsky}). \\ & \text{spouse}(\text{chelsea_clinton}, \text{marc_mezvinsky}). \\ & \text{is_a_congressman}(\text{edward_mezvinsky}). \\ & \text{father_in_law}(A, B) \leftarrow \text{spouse}(B, C), \text{child_of}(C, A). \end{aligned}$$

meaning that Chelsea is the child of Bill Clinton, Marc Mezvinsky is the child of Edward Mezvinsky, Chelsea is the spouse of Marc, Edward Mezvinsky is a congressman and A is father in law of B when the spouse of B is a child of A.

The inference over Π will not materialize the answer $X = \text{chelsea_clinton}$ and $Y = \text{edward_mezvinsky}$, because despite the statement and the rule describing the same sub-domain, there is no precise vocabulary matching between the query and Π .

In order for the reasoning to work, the approximation of the following terms would need to be established: $\text{daughter_of} \sim \text{child_of}$, $\text{is_a_congressman} \sim \text{politician}$. The reasoner should be able to semantically approximate vocabulary terms such as *daughter_of* and *child_of*, addressing the terminological gap required by this inference.

To close the semantic/vocabulary gap in a traditional deductive logic knowledge base it would be necessary to increase the size of Π to such an extent that it would contain all the facts and rules necessary to cope with any potential vocabulary difference. Together with the aggravation of the scalability problem, it would be necessary to provide a principled mechanism to build such a large scale and consistent set of facts and rules.

10.4.3 Distributional Logic Programs

Definition 10.2. Let Π_1 and Π_2 be logic programs with signatures, resp., $\Sigma_{\Pi_1} = (P_{\Pi_1}, E_{\Pi_1})$ and $\Sigma_{\Pi_2} = (P_{\Pi_2}, E_{\Pi_2})$. We say that Π_1 and Π_2 are semantically related, taking into account a threshold η (or sr-logic programs wrt η) where there is some predicate substitution $\lambda_\eta(P_1, P_2)$ such that $\Pi_2 = \Pi_1 \cdot \lambda_\eta(P_1, P_2)$ where $P_1 = (P_{\Pi_1} \setminus P_{\Pi_2})$ and $P_2 = (P_{\Pi_2} \setminus P_{\Pi_1})$.

Definition 10.2 states that two *sr-logic programs* are different versions of the same program that use a set of different predicate symbols, which are semantically related wrt a DSM. From the logical point of view, the answer set models of Π_1 are preserved in Π_2 (and vice-versa) in the sense that the extensions of all predicates in both programs are the same: different predicate symbols that are semantically related have the same extension:

Proposition 10.3. Let Π be a normal logic program, $S \subseteq HB_\Pi$ be a set of atoms. For any predicate substitution λ_η , $(\Pi^S \cdot \lambda_\eta) = (\Pi \cdot \lambda_\eta)^{S \cdot \lambda_\eta}$.

Corollary 10.4. Let Π_1 and Π_2 be sr-logic programs wrt η and S a set of atoms such that $P_S \subseteq P_{\Pi_1}$. Then $\Pi_1^S = (\Pi_2^{S \cdot \lambda_\eta(P_1, P_2)}) \cdot \lambda_\eta(P_2, P_1)$.

The semantic relatedness sr_{prog} between logic programs Π_1 and Π_2 and the semantic relatedness sr_{models} between (answer set) models $\mathcal{M}(\Pi_1)$ and $\mathcal{M}(\Pi_2) = \mathcal{M}(\Pi_1) \cdot \lambda_\eta(P_1, P_2)$ are defined using the predicate substitution $\lambda_\eta(P_1, P_2)$ used to transform Π_1 in Π_2 :
 $sr_{prog}(\Pi_1, \Pi_2) = sr_{models}(\mathcal{M}(\Pi_1), \mathcal{M}(\Pi_2)) = sr_{subst}(\lambda_\eta(P_1, P_2))$

Algorithm 13 summarizes the distributional predicate substitution algorithm for logic programs.

An example of the running algorithm is described below.

Let Π be formed by:

$$\begin{aligned} & child_of(chelsea_clinton, bill_clinton). \\ & child_of(marc_mezvinsky, edward_mezvinsky). \\ & spouse(chelsea_clinton, marc_mezvinsky). \\ & is_a_congressman(edward_mezvinsky). \\ & father_in_law(A, B) \leftarrow spouse(B, C), child_of(C, A). \end{aligned}$$

Suppose that we want to answer the query “*Is the father in law of Bill Clinton’s daughter a politician?*” for a threshold $\eta = 0.05$:

Algorithm 13 Distributional Predicate Substitution Algorithm - DPS**INPUT**

- P_{Π} : The list of all predicate symbols that appear in a program Π
- P_{query} : The list of all predicate symbols q that appear in a query Q such that $q \notin P_{\Pi}$
- η : Threshold

OUTPUT

- *Substitutions*: A set with all predicate substitutions $\lambda_{\eta}(P_{query}, P'_{\Pi})$ where $P'_{\Pi} \subseteq P_{\Pi}$ and $|P'_{\Pi}| = |P_{query}|$

PROCEDURE $DPS(P_{\Pi}, P_{query}, \eta)$:

```

1: if  $P_{query} == []$  then
2:   return ([ [] ])
3: else
4:   for all  $i \in [1, |P_{query}|]$  do
5:      $X \leftarrow P_{query}(i)$ 
6:      $P'_{query} \leftarrow remove(X, P_{query})$ 
7:     Substitutions  $\leftarrow []$ 
8:     for all  $Y \in P_{\Pi}$  do
9:       if  $sr(X, Y) > \eta$  then
10:         $P'_{\Pi} \leftarrow remove(Y, P_{\Pi})$ 
11:        Subst  $\leftarrow []$ 
12:        for all  $Z \in DPS(P'_{\Pi}, P'_{query}, \eta)$  do
13:          Subst  $\leftarrow append(Z, [(X, Y, sr(X, Y))])$ 
14:          Substitutions  $\leftarrow append(Substitution, [Subst])$ 
15:        end for
16:      end if
17:    end for
18:  end for
19: end if
20: return Substitutions

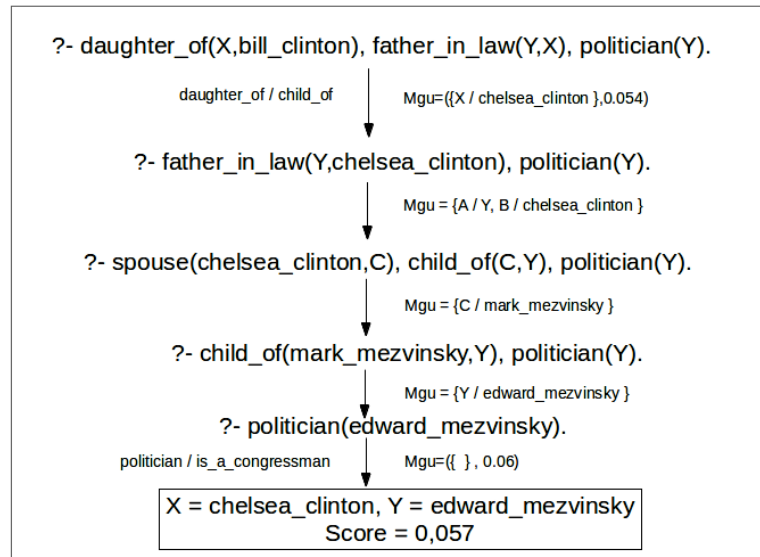
```

?-((daughter_of(X, bill_clinton), father_in_law(Y, X), politician(Y)), 0.05).

Since the predicate *daughter_of* does not appear in Σ_{Π} , we need to verify if there is a semantically related binary predicate to *daughter_of* for $\eta = 0.05$. As can be seen in table 10.8, only *child_of* is semantically related to *daughter_of* wrt η ($sr(child_of, daughter_of) = 0.054 > 0.05$). Thus, we allow that these predicates unify and they have a *mgu* ($\{X/chelsea_clinton\}, 0.054$). The complete inference is shown in figure 10.8 and the score of the answer is $(0.054 + 0.06)/2 = 0.057$.

TABLE 10.8: Semantic relatedness determined by the τ -Space module between the predicates in Q and Π , according to arity.

sr	<i>child_of</i> /2	<i>spouse</i> /2	<i>father_in_law</i> /2	<i>is_a_congressman</i> /1
<i>daughter_of</i> /2	0.054	0.012	0.048	-
<i>politician</i> /1	-	-	-	0.06

FIGURE 10.8: Derivation for the question ‘*Is the father in law of Bill Clinton’s daughter a politician?*’

10.5 Related work on the interface between structured data, logics and distributional semantics

Different works have previously applied distributional semantic models to structured/logical knowledge bases. Previous models had different application scenarios and supporting distributional models. In this section, these works are briefly described.

Speer et al. (2008) [248] introduced AnalogySpace, a hybrid distributional-relational model over ConceptNet using Latent Semantic Indexing. Cohen et al.(2009) [249] proposes PSI, a distributional model that encodes predications produced by the SemRep system. The τ -Space distributional-relational model is similar to AnalogySpace and PSI. Differences in relation to these works are: (i) the supporting distributional model (τ -Space is based on Explicit Semantic Analysis), (ii) the use of the reference corpus (the τ -Space distributional model uses an independent large scale text corpora to build the distributional space, while PSI builds the distributional model based on the indexed

triples), (iii) the application scenario (the τ -Space is evaluated under an open domain scenario while PSI is evaluated on the biomedical domain), (iv) the focus on evaluating the selectivity and ability to cope with incompleteness. Cohen et al.(2012) extends the discussion on the PSI to search over triple predicate pathways in a database of predications extracted from the biomedical literature by the SemRep system. Taking the data as a reference corpus, Novacek et al.(2011) [250] build a distributional model which uses a PMI-based measure over the triple corpora. The approach was evaluated using biomedical semantic web data.

In [251], Novacek et al. (2010), proposes the application of emergent knowledge embedded in text to enrich asserted publication metadata knowledge in the design of a search & browse over publications metadata.

In [252], Lukasiewicz & Straccia presented probabilistic fuzzy dl-programs, which is a uniform framework that deals with uncertainty and fuzzy vagueness. This work focus on the ontology mapping aspect (uncertainty) and in the use of a distributional semantic approach to align semantically equivalent terms. The common goal of both fuzzy/probabilistic and distributional approaches is the introduction of flexibility into the reasoning process. The main benefit of using distributional semantics is the use of large-scale unstructured or semi-structured information sources to complement the semantics of logic programs. One of the strengths of distributional semantic models is from the acquisitional perspective, where comprehensive semantic models can be automatically built from large-scale corpora.

Distributional semantic models are evolving in the direction of coping with better compositional principles, supporting the semantic interpretation of complex sentences/statements. Baroni et al. [155] provide an extensive discussion of state of the art approaches for compositional-distributional models. In this work the compositional model is given by the structure of the logical atoms in a logic program Π , which defines a set of vectors in the distributional vector space.

In [253], Grefenstette presented how elements of a quantifier-free predicate calculus can be modeled using tensors and tensor contraction. The basic elements, truth values and domains objects, are modelled as vectors and predicates and relations are modeled through high order tensors. Also, Boolean connectives are modeled using tensors and with the basic elements used to build a quantifier-free predicate calculus.

10.6 Hybrid Distributional-Relational Models (DRMs)

10.6.1 Types of Distributional-Relational Models

Previous works have started to explore the connection between distributional semantics and structured models. This section aims at positioning this work against existing models, providing a schematic synthesis of distributional-relational models (DRMs).

DRMs support a double perspective of semantics, keeping the fine-grained precise semantics of the structured KB but also complementing it with the distributional model. Two main categories of DRMs and associated applications can be distinguished: *semantic matching*, which is the target of this work and *knowledge discovery*.

10.6.1.1 Semantic Matching

In this category the reference corpus (\mathcal{RC}) is typically unstructured and it is distinct from the \mathcal{KB} . The large-scale *unstructured* \mathcal{RC} is used as a *commonsense knowledge base*.

This work is positioned in this category, where the DRM ($\tau - Space$) is used for supporting schema-agnostic queries over the structured \mathcal{KB} : terms used in the query are projected into the distributional vector space and are semantically matched with terms in the \mathcal{KB} via distributional semantics using commonsense information embedded on large scale unstructured corpora \mathcal{RC} .

The reasoning application scenario described in Section 10.3 is also in this category, where the $\tau - Space$ to support selective reasoning over commonsense $\mathcal{KB}s$. Distributional semantics is used to select the facts which are semantically relevant under a specific reasoning context, allowing the scoping of the reasoning context and also coping with incomplete knowledge of commonsense KBs .

Similarly, the scenario described in Section 10.4 used the $\tau - Space$ to support approximate reasoning on logic programs by defining predicate substitutions.

The work of Novacek et al. on the CORAAL search engine [251] is also positioned under this category, where publication metadata is enriched with emergent knowledge from a corpus of publication texts.

10.6.1.2 Knowledge Discovery

In this category, the structured \mathcal{KB} is used as a distributional reference corpus (where $\mathcal{RC} = \mathcal{KB}$). Implicit and explicit semantic associations are used to derive new meaning and discover new knowledge. The use of structured data as a distributional reference corpus is a pattern used for knowledge discovery applications, where knowledge emerging from *similarity patterns in the data* can be used to retrieve similar entities and expose implicit associations. In this context, the ability to represent the \mathcal{KB} entities' attributes in a vector space and the use of vector similarity measures as way to retrieve and compare similar entities can define universal mechanisms for knowledge discovery and semantic approximation.

Novacek et al. [250] describe an approach for using web data as a bottom-up phenomena, capturing meaning that is not associated with explicit semantic descriptions, applying it to entity consolidation in the life sciences domain. Speer et al. [248] proposed AnalogySpace, a DRM over a commonsense \mathcal{KB} using Latent Semantic Indexing targeting the creation of the analogical closure of a semantic network using dimensional reduction. AnalogySpace was used to reduce the sparseness of the \mathcal{KB} , generalizing its knowledge, allowing users to explore implicit associations. Cohen et al. [249] introduced PSI, a predication-based semantic indexing for biomedical data. PSI was used for similarity-based retrieval and detection of implicit associations.

10.6.2 The Distributional Data Stack

DRMs provide universal mechanisms which have fundamental features for semantic systems:

- *Built-in semantic approximation for terminological and instance data;*
- *Ability to use large-scale unstructured data as commonsense knowledge;*
- *Ability to detect emerging implicit associations in the \mathcal{KB} with the computation of vector-space based similarity models;*
- *Simplicity of use supported by the vector space model abstraction;*
- *Robustness with regard to poorly structured, heterogeneous and incomplete data;*

These features provide a framework for a robust and easy-to-deploy semantic approximation component grounded on large-scale data. Considering the relevance of these

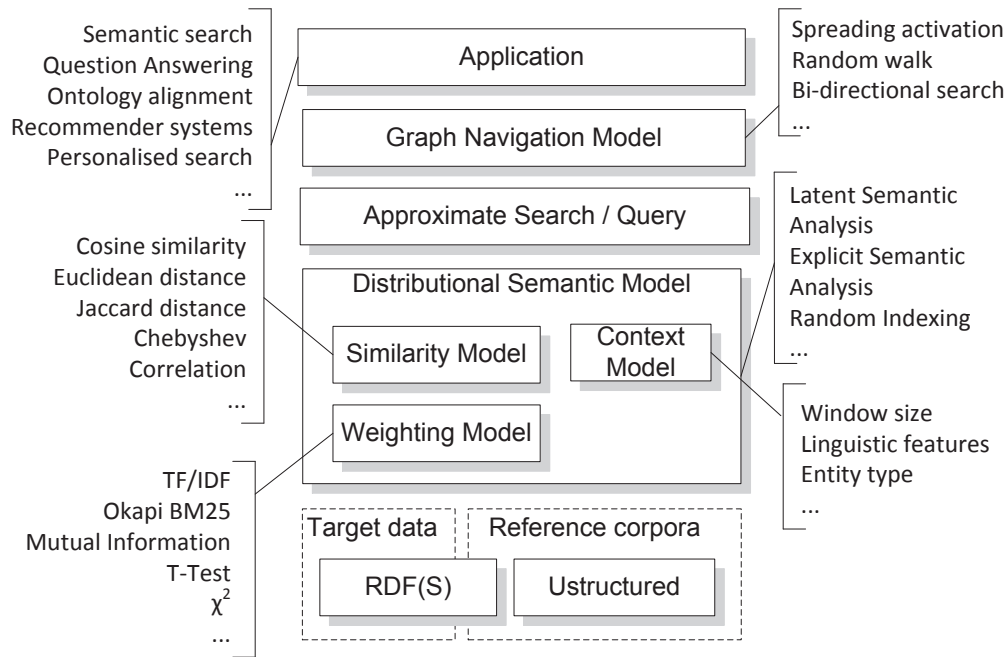


FIGURE 10.9: Distributional Data stack.

features in the deployment of semantic systems in general, this work synthesizes its vision by proposing a *Distributional Data stack* abstraction (Figure 10.9).

At the bottom of the stack, unstructured and structured data can be used as reference corpora together with the target \mathcal{KB} (RDF(S)). Different elements of the distributional model are included as optional and composable elements of the architecture. The *approximate search and query operations layer* access the *DSM layer*, supporting users with semantically flexible search and query operations. A *graph navigation layer* defines graph navigation algorithms (e.g. such as spreading activation, bi-directional search) using the semantic approximation and the distributional information from the layers below.

10.7 Chapter Summary

This chapter describes two application scenarios of the hybrid distributional-relational model and its semantic approximation mechanism. The first scenario focuses on the development of *schema-agnostic logic programs*, while the second scenario focused on *selective commonsense reasoning on incomplete knowledge bases*. Additionally, the proposed model is generalized into a knowledge-based semantic interpretation (KBSI) model and into a reference architecture as a Distributional Data Stack. Associated publications to this chapter are [177, 254, 255, 256, 257].

Chapter 11

Conclusion

11.1 Thesis Summary

As databases and data environments grow in size and heterogeneity, and as structured data becomes reused outside its original creation context (*open communication* scenarios), mechanisms to support users interacting, querying and exploring data without the needs to understand its specific representation lexicon and structure becomes a fundamental demand for contemporary data management.

Chapter 2 concentrated on the analysis of the changes in the database landscape, motivating how the growth in size, complexity, dynamicity and decentralisation of schemas (SCoDD) are bringing fundamental changes in data management. These changes strongly impact the effectiveness of existing approaches for querying structured data. At the center of this problem is the concept of semantic heterogeneity between query and databases. The dimensions and causes of semantic heterogeneity were analysed under the contemporary data management perspective.

While the understanding of the motivation for schema-agnostic queries is progressively becoming a known concern, there is a lack of categorization to express different semantic challenges that a schema-agnostic query mechanism need to cope with. In order to address this gap, this work provides a classification based on semantic mapping categories, which define the *semantic resolvability* of a query, i.e. the level of complexity involved in mapping a query to a database. The goal is to provide an initial classification framework which can be refined collectively and which could both help in the understanding of the challenges of schema-agnostic queries and on the scoping in the evaluation of exiting approaches.

Chapter 3 analyses the state-of-the-art for querying and searching structured data with regard to schema-agnosticism. Different categories of approaches including Natural Language Interfaces, Flexible Query Interfaces and Semantic Search over RDF are analysed relative to the set of core requirements. Associated publications to this chapter are [143, 144].

At the center of schema-agnostic query mechanisms is the definition of a semantic model which can cope with the *semantic resolvability categories*. Chapter 4 provides an analysis of the semiotic principles behind human-database communication and the associated semantic perspective on databases. Different perspectives on semantics (logical, cognitivist and structuralist) are analysed. Based on the analysis, a hybrid *distributional-relational semantic model* is outlined targeting to address the new semiotic assumptions which emerge in the *open communication scenario*. The associated publication to this chapter is [177].

Chapter 5 provided a *quantitative information-theoretic analysis* of the semantic complexity associated with matching schema-agnostic queries. The core goal of the chapter is to provide a *quantitative model* for schema-agnostic query-database matching. Different entropy measures corresponding to different variables are defined, and approximative models are proposed when exact models are not feasible to be calculated. The analysis of the entropy measures indicate a substantial reduction of the matching entropy with the use of a *semantic pivot-based model*, in which elements with lower semantic matching entropies are resolved first, providing a context-based reduction mechanism of the entropy values for the remaining mappings. The associated publications to this chapter are [121, 183].

Chapter 6 formalized the definition of the τ – *Space*, a semantic representation for the *hybrid distributional-relational model*. The τ – *Space* is a vector space model emerging from the embedding of a data graph into a distributional semantics vector space. At the τ – *Space*, each element in the data graph has an associated distributional semantics vector representation, which supports a geometric-based semantic approximation model, using the distributional knowledge on a large-scale reference corpora. The structure of the τ – *Space* is defined by the mapping between data model categories and the associated distributional subspaces associated with each category. Associated publications to this chapter are [176, 198, 199, 200, 201, 202, 203].

Chapter 7 described the distributional semantic search approach in which the *distributional semantic relatedness measure* is used as a *ranking function*. The *semantic differential* approach for the determination of the semantic relatedness-based ranking score is introduced, supporting the filtering of unrelated results. The semantic search is evaluated for an open domain terminology-level search scenario, achieving superior query

coverage when compared to WordNet-based query expansion. Associated publications to this chapter are [222, 223].

Using the $\tau - Space$ as a semantic representation approach, the semantic search and the entropy minimization proposed in the previous chapters, a *schema-agnostic query processing approach* is described in Chapter 8. The query processing approach uses a set of *semantic search, composition and data transformation operations* over the $\tau - Space$. A supporting architecture for the query mechanism is proposed. The architecture is instantiated into the *Treo* prototype, a schema-agnostic natural language query mechanism. Associated publications to this chapter are [198, 203, 235, 236, 237, 238, 239, 240, 241, 242, 243].

The proposed schema-agnostic query approach is evaluated in Chapter 9 using the *Question Answering over Linked Data (QALD 2011)* test collection. The suitability of the test collection to support the evaluation of schema-agnostic queries is verified by statistically analyzing features of the test collection related to the thesis hypotheses. The query approach is evaluated using metrics which map to the set of core requirements for schema-agnostic queries. The proposed approach, confirmed the research hypotheses and had a high coverage of the core requirements for schema-agnostic queries under a semantic best-effort scenario (*high-recall* and *medium precision*). The *post-mortem* analysis of the query mechanism shows that limitations of the approach were concentrated on the transformation of natural language queries to the query plan. The associated publication to this chapter is [236].

The thesis concludes with the analysis of two application scenarios of the hybrid distributional-relational model and its semantic approximation mechanism (Chapter 10). The first scenario focuses on the development of *schema-agnostic logic programs*, while the second scenario focused on *selective commonsense reasoning on incomplete knowledge bases*. Additionally, the proposed model is generalized into a knowledge-based semantic interpretation (KBSI) model and into a reference architecture as a *Distributional Data Stack*. Associated publications to this chapter are [177, 254, 255, 256, 257].

11.2 Conclusions

In this section the hypotheses are analyzed and the associated conclusions are drawn, under the perspective of the evaluation.

Hypothesis I: Distributional semantics provides an accurate (I.1), comprehensive (I.2) and low maintainability (I.3) approach to cope with the abstraction-level and

lexical-level dimensions (**I.4**) of semantic heterogeneity (**A.1**) in schema-agnostic queries over large-schema open domain datasets (**A.2**).

– **Sub-Hypothesis I.1:** *accurate*

* **Experimental Support:** Mean Avg. Precision = 0.539, Avg. MRR = 0.431. Equivalent MAP to the best performing baseline system in precision.

* **Interpretation:** The proposed model provides a query approach with *medium-high accuracy*, where on average, the results are listed between second and third rank positions. The proposed model is effective for a semantic best-effort scenario, where there is no expectation of absolute precision (in contrast to structured database queries). The evaluation did not limit the number of returned results (i.e. did not constraint the precision measurement to the top-k results). This implies that the potential effort on users to filter out unrelated results is potentially low. Additionally, the *semantic best-effort scenario* is dependent on the provision of contextual mechanisms for users to interpret the correctness of the result-set. The evaluation of the filtering and interpretation effort under the semantic best-effort query scenario were left outside the scope of this thesis. The investigation of the queries with lower precision had as main causes query pre-processing errors (mapping the natural language to structured query candidates), and the introduction of false positives by the distributional semantics approximation model.

– **Sub-Hypothesis I.2:** *comprehensive*

* **Experimental Support:** Avg. Recall = 0.775, % of queries answered = 0.836, % of queries fully answered = 0.627, % of queries partially answered = 0.208. 16% improvement in recall relation to the best performing baseline systems (in recall) and 32 % improvement in % of queries answered in relation to the best performing system (in % of queries answered).

* **Interpretation:** The proposed model provides a query approach with *high recall* and *high percentage of queries answered* showing a significant improvement in relation to existing baseline systems. This confirms distributional semantic relatedness as a *comprehensive semantic approximation* method. Most errors affecting recall are related to query pre-processing (mapping natural language to structured query candidates).

– **Sub-Hypothesis I.3:** *low maintainability*

- * **Experimental Support:** Dataset specific a priori adaptation effort (minutes) = 0.00, Dataset specific semantic enrichment effort per query (secs) = 0.00, Dataset specific semantic disambiguation/filtering effort per query (secs) = 2.20.

- * **Interpretation:** The proposed query approach supports a low adaptation effort schema-agnostic query mechanism, with no effort involved at the dataset indexing time for semantic enrichment. Distributional semantics provides a semantic approximation mechanism which automatically extract meaning representations from large-scale corpora, not requiring dataset manual curation or intervention, reducing the effort involved in the maintainability and transportability of the approach. Users can interact with disambiguation dialogs to confirm correct semantic pivot and predicate alignments. The disambiguation mechanism serves as a precision improvement mechanism and does not affect the other metrics as it is dependent on the a priori distributional semantics alignment.

- **Sub-Hypothesis I.4:** *abstraction-level and lexical-level*

- * **Experimental Support:** QALD 2011 test collection: Mean Avg. Precision = 0.539, Avg. MRR = 0.431, Avg. Recall = 0.775. Vocabulary search: ESA = 92.25%, String matching = 45.77%, WordNet QE = 52.48%. 16 % improvement in recall relation to the best performing baseline system (in recall) and 32 % improvement in % of queries answered in relation to the best performing baseline system (in % of queries answered). Equivalent MAP to the best performing baseline system in precision.

- * **Interpretation:** Distributional semantics supports the alignments between query and dataset in an open domain scenario, providing a feasible semantic matching mechanism. However, the distributional mechanisms needs to be supported by a contextualization method (semantic pivoting) which reduces the search space and the semantic entropy associated with the matching process. The distributional semantic model for the query scenario was able to capture both alignments which represent syntagmatic and paradigmatic relations, alignments from terms from different lexical categories and different levels of abstraction. The high recall for the query test collection, higher recall in comparison to the best performing baseline system and the higher percentage in the number of alignments resolved for the vocabulary search scenario (in comparison with WordNet QE), shows that distributional semantics provides a comprehensive semantic approximation/matching mechanism. The precision and mean reciprocal rank values shows that distributional semantics provide a low number of false positive alignments, which need to be filtered out in its application scenarios.

- **Assumption A.1:** *semantic heterogeneity*

* **Evidential Support:** number of editors $> 10^3$ s, 10^6 s.

* **Interpretation:** The evaluation dataset (DBpedia) is derived from structured and semi-structured information present in Wikipedia.

Wikipedia has a large number of active editors, estimated in more than $> 10^3$ s, 10^6 s. While DBpedia organize part of its terminology-level data into an ontology, most of the properties and classes present in the dataset are outside DBpedia ontology. Consequently, DBpedia is a decentralized and collaboratively created dataset, defining a semantically heterogeneous conceptual model. Additionally, DBpedia has an intrinsic semantic heterogeneity due to the comprehensive domain coverage.

– **Assumption A.2:** *large-schema open domain datasets*

* **Evidential Support:** # of classes, and properties $> 10^4 - 10^5$ s, # of records (triples) $> 10^6$ s.

* **Interpretation:** The number of terminology-level elements (# of classes and properties) shows that the evaluation dataset (DBpedia) supports the evaluation of the approach under a large-schema scenario. The # of records supports the evaluation under a large number of facts and instances, providing a large dataset from the schema-agnostic perspective. The evaluation is, however, limited to one comprehensive dataset.

Conclusion (Hypothesis I): The evaluation provides sufficient corroboration evidence to support Hypothesis I.

The hypothesis can be re-written into a more precise form taking into account the evaluation findings:

Hypothesis I (Reformulated): Distributional semantics provides a *high-recall, medium-high precision* and low maintainability approach to cope with the abstraction-level and lexical-level dimensions of semantic heterogeneity in schema-agnostic queries over large-schema open domain datasets.

Hypothesis II: The compositional semantic model defined by the query planning mechanism supports expressive **(II.1)** schema-agnostic queries over large-schema open domain datasets **(A.2)**.

* **Sub-Hypothesis II.1:** *expressive*

· **Evidential Support:** Avg. Recall = 0.775, % of queries answered = 0.836, % of queries fully answered = 0.627, % of queries partially answered = 0.208. 16 %

improvement in recall relation to the best performing baseline systems (in recall) and 32 % improvement in % of queries answered in relation to the best performing baseline system (in % of queries answered).

- **Interpretation:** The compositional semantic model defined by the query planning mechanism provides a comprehensive generalization of query-dataset structural matching patterns, which is confirmed by the high recall, high % of queries answered and the improvements over the baselines in both dimensions. The proposed compositional model focuses on a semantic interpretation model which defines the role of context in the query-dataset alignment (reducing the impact on ambiguity, vagueness and synonyms) also making more explicit the role of conceptual approximation in this process. The number of distinct query patterns present in the test collection combined with the distinct categories of query-dataset lexico-conceptual differences provides a comprehensive evaluation set-up for basic factoid queries. It is likely that the number of possible query patterns follows a long-tail distribution and that the evaluation presented in this work explored the set of most frequent query patterns. A systematic study of existing query patterns using a more comprehensive query set can provide a fundamental resource for the interpretation of the coverage of test collections and it is indicated as future work.

Conclusion (Hypothesis II): The evaluation provides sufficient corroboration evidence to support Hypothesis II.

Hypothesis III: The proposed distributional-relational structured vector space model (τ - *Space*) supports the development of a schema-agnostic query mechanism with interactive query execution time (**III.1**), low index construction time (**III.2**) and size (**III.3**) and scalable (**III.4**) to large-schema open domain datasets (**I.7**).

* **Sub-Hypothesis III.1:** *interactive query execution time*

- **Evidential Support:** Avg. query execution time (ms) = 8,530, shortest query execution time (ms) \leq 2,000 ms.
- **Interpretation:** Most of the queries have query execution time below 2,000 ms using a single commodity configuration computer (Intel iCore5 8GB RAM) to host the query engine. Queries which take longer (60,000 ms) are due to the number of records and the application of operations such as conditional filters (which are not optimized in the prototype query engine). The cost of the semantic approximation operations ($<$ 1,000 ms) supports an interactive query execution time query mechanism.

* **Sub-Hypothesis III.2:** *low index construction time*

- **Evidential Support:** Avg. index insert time per triple (ms) = 5.35.
- **Interpretation:** The creation of a distributional semantic index has the intrinsic cost of requesting and indexing the distributional vector associated with a term which is indexed for the first time. As repeated terms are indexed, the cost associated with the distributional semantics component is eliminated. At the beginning of the indexing process as new terms appear more frequently the index cost is larger. As the probability occurrence of new terminology-level becomes more rare along the indexing process, the temporal overhead of the distributional indexing tends to zero.

* **Sub-Hypothesis III.3:** *low index size*

- **Evidential Support:** Avg. index size per triple (bytes) = 250. Index/dataset ratio = 1.2 (with the index also working also as a triple store: 0.2 purely the distributional index).
- **Interpretation:** The distributional semantic index has a medium index size overhead associated with the indexing of the distributional context vectors.

* **Sub-Hypothesis III.4:** *scalable*

- **Evidential Support:** Avg. query execution time (ms) = 8,530, shortest query execution time (ms) \leq 2,000. Avg. index insert time per triple (ms) = 5.35. Avg. index size per triple (bytes) = 250. Index/dataset ratio = 1.2 (dataset working also as a triple store).
- **Interpretation:** Distributional semantics introduces a low impact overhead in terms of query execution time and a medium impact on indexing time and index size. From the indexing perspective, as the dataset grows, and the probability of terms which were not present before decreases, the impact on indexing time and size tends to drastically decrease. From the perspective of the query execution time, there is evidence that the approach scales, supported by the segmentation and reduction of the search space by the semantic pivot. Additionally, both the semantic pivot and the distributional vector space supports the segmentation of the index for a distributed indexing/search. However, the empirical corroboration of scalability was left outside the scope of this work.

Conclusion (Hypothesis III): The evaluation provides sufficient corroboration evidence to support Hypothesis III.

11.3 Limitations & Open Questions

Different dimensions were left outside the scope of this work. Below the most relevant dimensions are described:

- * **Suitability of distributional semantic models for domain specific datasets:** The evaluation on this work focused on open domain scenarios. The suitability of distributional semantics for domain specific scenarios (e.g. biological, financial datasets) was not evaluated in the scope of this work.
- * **Evaluation over multiple datasets:** The approach was evaluated for a single large-scale heterogeneous dataset. The suitability of the proposed approach to query multiple datasets was not verified.
- * **Scalability evaluation:** While the performance indicators provide some level of support for the analysis of the scalability of the approach, no specific scalability evaluation was performed.

11.4 Main Contributions

This thesis focused on the proposal of a schema-agnostic query for large-schema/schema-less heterogeneous databases. At the center of this model is the proposal of a semantic model which can support an efficient semantic matching model for schema-agnostic queries in an open communication scenario. The proposed semantic model uses distributional semantic models at its center, which, aligned with the context provided by the structured data, facilitates the semantic approximation process and the alignment between query and data. While the proposal and evaluation of a distributional semantics-based schema-agnostic query approach is at the core of this thesis, the thesis also targeted the development of a more ample discussion on the motivation, principles and applications of schema-agnostic queries and distributional-based semantic approximation mechanisms.

The items below summarizes the contributions of this thesis:

- * **Definition of a preliminary model for mapping schema-agnostic queries**
- * **Definition of a preliminary information-theoretical semantic complexity model for schema-agnostic queries**

- * Definition of the schema-agnostic query processing approach based on distributional semantics
- * Evaluation of the suitability of distributional semantics for the conceptual mapping between queries and databases
- * Evaluation of the suitability of the approach for addressing schema-agnostic queries
- * Evaluation of the temporal performance for the proposed schema-agnostic approach (query execution time, indexing time)
- * Creation of a natural language interface(NLI)/question answering(QA) system over RDF(S) data
- * Creation of a basic distributional-relational infrastructure ($\tau - DB$)
- * Definition and creation of a distributional semantic index for supporting schema-agnostic queries
- * Analysis of the Question Answering over Linked Data 2011 test collection with regard to its ability to evaluate schema-agnostic queries
- * Extension of the evaluation methodology for Question Answering Systems over Linked Data to include temporal performance (query/indexing), index size and maintainability metrics
- * Discussion of the motivation for schema-agnostic queries
- * Generalization of the proposed approach as a semantic interpretation model
- * Discussion of two application scenarios for the proposed semantic approximation model

11.5 Future Research Directions

Preliminary results for distributional-relational models (DRMs) have been encouraging, showing the effectiveness of distributional semantics as a semantic approximation mechanism. The universality of the vocabulary problem and the demand for effective semantic approximation mechanisms, together with the simplicity and effectiveness of distributional semantics in addressing it, will motivate the further development of research on

the field. Important short-term research challenges which became evident during the elaboration of this work are:

* **Investigation of uncertainty models for distributional-relational models**

- *Improvement of the connection between distributional semantics, probability, fuzzy set and information theory:* Definition of the probabilistic and information-theoretic frameworks for DRMs which can support the modelling of uncertainty measures for distributional models.
- *Better selection of linguistic features:* Improving the understanding of uncertainty models can support the improvement of distributional semantic models by selecting features in the corpora which minimizes uncertainty for a specific dataset or application scenario.
- *Definition of soundness and completeness conditions for schema-agnostic queries:* Given a set of queries, a corpora and a dataset, verify the soundness and completeness conditions for the queries both from the conceptual and structural mapping perspectives.
- *Minimum evidence DSM-DRM models:* Given a dataset or a query set, define conditions to minimize the size of evidence set for a DSM.

* **Formalization of the distributional-relational algebra & query optimization approaches:** Exploration of the formal aspects of DRMs, including an extension of relational algebra and the modelling of different query optimization approaches or distributional semantics.

* **Analysis of the suitability of the distributional model for domain-specific semantic approximations:** Define the lexical, semantic and statistical properties that a domain-specific corpora should have to support a distributional model. Comparative analysis for the suitability of DSMs for different domains (e.g. biomedical, financial).

* **More comprehensive and systematic comparative study of distributional semantic models for open domain schema-agnostic queries:**

* **Software Infrastructures**

- *Better integration of distributional semantics software infrastructures to database platforms.*

-
- *Creation of robust and easy-to-use distributional semantics software infrastructures.*

 - * **Investigation of the impact of Distributional-DBMSs (D-DBMS) on information systems:** Investigation of how information systems are affected by a semantic abstraction layer over databases including general principles, components and architectures.

Appendix A

QALD 2011 Query Set

The list below provides the set of natural language queries and the corresponding SPARQL queries for the Question Answering over Linked Data (QALD) test collection.

```
<string>Give me all school types.</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago: <http://dbpedia.org/class/yago/>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type yago:SchoolTypes .
OPTIONAL {
?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
</query>
```

```
<string>Which presidents were born in 1945?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX yago: <http://dbpedia.org/class/yago/>
SELECT DISTINCT ?uri ?string WHERE {
{
?uri rdf:type onto:President .
?uri onto:birthDate ?date .
FILTER regex(?date, '^1945') .
OPTIONAL
{
```

```

?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
UNION {
?uri rdf:type yago:President.
?uri onto:birthDate ?date .
FILTER regex(?date, '^1945') .
OPTIONAL {
?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') } }
}
</query>

```

<string>Who are the presidents of the United States?</string>

```

<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prop: <http://dbpedia.org/property/>
SELECT DISTINCT ?uri ?string WHERE {
{ ?uri rdf:type yago:PresidentsOfTheUnitedStates. }
UNION
{ ?uri rdf:type onto:President. ?uri prop:title
res:President_of_the_United_States. }
OPTIONAL
{ ?uri rdfs:label ?string. FILTER (lang(?string) = 'en') }
}
</query>

```

<string>Who was the wife of President Lincoln?</string>

```

<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT ?uri ?string WHERE {
?person rdf:type onto:President .
?person foaf:surname 'Lincoln'@en .
?person onto:spouse ?uri.
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}

```

```
}
</query>

<string>Who developed the video game World of Warcraft?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT ?uri ?string WHERE {
?subject rdf:type onto:Software .
?subject rdfs:label 'World of Warcraft'@en .
?subject onto:developer ?uri .
OPTIONAL
{?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
</query>

<string>What is the official website of Tom Hanks?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?uri WHERE {
?subject rdfs:label 'Tom Hanks'@en .
?subject foaf:homepage ?uri
}
</query>

<string>
List all episodes of the first season of the HBO television series
The Sopranos!
</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT ?uri ?string WHERE {
?uri onto:series res:The_Sopranos .
?uri onto:seasonNumber 1 .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>
```

```
<string>Who produced the most films?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT ?uri ?string WHERE {
?film rdf:type onto:Film .
?film onto:producer ?uri .
OPTIONAL {
?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
ORDER BY DESC(COUNT(?film)) LIMIT 1
</query>

<string>Which people have as their given name Jimmy?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type foaf:Person.
?uri foaf:givenName 'Jimmy'@en .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

<string>Is there a video game called Battle Chess?</string>
<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> ASK WHERE {
?software rdf:type onto:Software .
?software rdfs:label ?name .
FILTER (regex(?name, 'Battle Chess'))
}
</query>

<string>Which mountains are higher than the Nanga Parbat?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```

PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?uri ?string WHERE {
?uri rdf:type onto:Mountain .
?acon rdfs:label 'Nanga Parbat'@en .
?acon prop:elevationM ?elevation .
?uri prop:elevationM ?allelevation .
FILTER (?allelevation > ?elevation) .
OPTIONAL {?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
</query>

```

```

<string>Who created English Wikipedia?</string>
<query>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?uri ?string WHERE {
?website rdf:type onto:Website .
?website onto:author ?uri .
?website rdfs:label 'English Wikipedia'@en .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

```

```

<string>Give me all actors starring in Batman Begins.</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT ?uri ?string WHERE {
?film rdf:type onto:Film .
?film onto:starring ?uri .
?film foaf:name 'Batman Begins'@en .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

```

```

<string>

```

Which software has been developed by organizations founded in California?

</string>

<query>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX onto: <http://dbpedia.org/ontology/>

PREFIX res: <http://dbpedia.org/resource/>

SELECT ?uri ?string WHERE {

?company rdf:type onto:Organisation .

?company onto:foundationPlace res:California .

?uri onto:developer ?company .

?uri rdf:type onto:Software .

OPTIONAL {

?uri rdfs:label ?string .

FILTER (lang(?string) = 'en') }

}

</query>

<string>

Which companies work in the aerospace industry as well as on nuclear reactor technology?

</string>

<query>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX onto: <http://dbpedia.org/ontology/>

PREFIX res: <http://dbpedia.org/resource/>

PREFIX prop: <http://dbpedia.org/property/>

SELECT ?uri ?string WHERE {

?uri rdf:type onto:Company .

?uri prop:industry res:Aerospace .

?uri prop:industry res:Nuclear_reactor_technology .

OPTIONAL

{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }

}

</query>

<string>Is Christian Bale starring in Batman Begins?</string>

<query>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

PREFIX onto: <http://dbpedia.org/ontology/> ASK WHERE {

?film rdf:type onto:Film .

?film onto:starring ?actors .

```
?actors rdfs:label 'Christian Bale'@en .
?film foaf:name 'Batman Begins'@en
}
</query>
```

```
<string>Is Christian Bale starring in Batman Begins?</string>
<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/> ASK WHERE {
?film rdf:type onto:Film .
?film onto:starring ?actors .
?actors rdfs:label 'Christian Bale'@en .
?film foaf:name 'Batman Begins'@en
}
</query>
```

```
<string>
Give me the websites of companies with more than 500000 employees.
</string>
<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX prop: <http://dbpedia.org/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri WHERE {
?subject rdf:type onto:Company .
?subject prop:numEmployees ?employees .
FILTER( xsd:integer(?employees) >= 500000 ) .
?subject foaf:homepage ?uri .
}
</query>
```

```
<string>Which actors were born in Germany?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX yago: <http://dbpedia.org/class/yago/>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type onto:Actor .
```

```
{ ?uri onto:birthPlace res:Germany . }
UNION
{ ?uri onto:birthPlace ?city . ?city rdf:type yago:StatesOfGermany . }
OPTIONAL {
?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
</query>
```

```
<string>Which country does the Airedale Terrier come from?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?string WHERE {
?dog rdfs:label 'Airedale Terrier'@en .
?dog prop:country ?string
}
</query>
```

```
<string>Which birds are there in the United States?</string>
<query>
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type yago:BirdsOfTheUnitedStates.
OPTIONAL
{?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
</query>
```

```
<string>Give me all European Capitals!</string>
<query>
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?uri ?string WHERE {
?uri rdf:type yago:CapitalsInEurope.
OPTIONAL {
?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>
```



```
<string>Which cities have more than 2 million inhabitants?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX prop: <http://dbpedia.org/property/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type onto:City.
{ ?uri prop:population ?population. }
UNION
{ ?uri prop:populationUrban ?population. }
FILTER (xsd:integer(?population) > 2000000) .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>
```

```
<string>Who was Tom Hanks married to?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prop: <http://dbpedia.org/property/>
SELECT DISTINCT ?uri ?string WHERE {
?person rdfs:label 'Tom Hanks'@en .
?person prop:spouse ?string .
OPTIONAL { ?uri rdfs:label ?string . }
}
</query>
```

```
<string>When was DBpedia released?</string>
<query>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?date WHERE {
?website rdf:type onto:Software .
?website onto:releaseDate ?date .
?website rdfs:label 'DBpedia'@en
}
</query>
```

```
<string>When was DBpedia released?</string>
```

```
<query>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?date WHERE {
?website rdf:type onto:Software .
?website onto:releaseDate ?date .
?website rdfs:label 'DBpedia'@en
}
</query>
```

```
<string>Which people were born in Heraklion?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX yago: <http://dbpedia.org/class/yago/>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type onto:Person .
?uri onto:birthPlace ?city .
?city rdfs:label 'Heraklion'@en
OPTIONAL
{?uri rdfs:label ?string .
FILTER (lang(?string) = 'en')}
}
</query>
```

```
<string>Which caves have more than 3 entrances?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?uri ?string WHERE {
?uri rdf:type onto:Cave .
?uri onto:numberOfEntrances ?entrance .
FILTER (?entrance > 3) .
OPTIONAL
{?uri rdfs:label ?string .
FILTER (lang(?string) = 'en')}
}
</query>
```

```
<string>Give me all films produced by Hal Roach.</string>
```

```

<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type onto:Film .
?uri onto:producer ?producer .
?producer rdfs:label 'Hal Roach'@en .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

```

```

<string>
Which software has been published by Mean Hamster Software?
</string>

```

```

<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX prop: <http://dbpedia.org/property/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT DISTINCT ?uri ?string WHERE {
?uri rdf:type onto:Software .
{ ?uri prop:publisher 'Mean Hamster Software'@en . }
UNION
{ ?uri onto:publisher res:Mean_Hamster_Software . }
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

```

```

<string>What languages are spoken in Estonia?</string>

```

```

<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri ?string WHERE {
?country rdf:type onto:Country.
{ ?country onto:language ?uri . }
UNION { ?uri onto:spokenIn ?country . }
FILTER (regex(?country, 'Estonia')).
}

```

```

OPTIONAL {?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
</query>

```

```

<string>Who owns Aldi?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT ?uri ?string WHERE {
?orga rdf:type onto:Organisation .
?orga onto:keyPerson ?uri .
?orga rdfs:label 'Aldi'@en .
OPTIONAL
{?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
FILTER (lang(?string) = 'en') }
</query>

```

```

<string>
Which capitals in Europe were host cities of the summer olympic games?
</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago: <http://dbpedia.org/class/yago/>
SELECT ?uri ?string WHERE {
?uri rdf:type yago:CapitalsInEurope .
?uri rdf:type yago:HostCitiesOfTheSummerOlympicGames .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

```

```

<string>
Who has been the 5th president of the United States of America?
</string>
<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?uri ?string WHERE {

```

```

?uri rdf:type onto:President .
?uri onto:orderInOffice '5th President of the United States'@en .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

<string>Who is called Dana?</string>
<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT DISTINCT ?uri ?string WHERE {
{ ?uri rdf:type foaf:Person. ?uri foaf:givenName 'Dana'@en. }
UNION
{?uri prop:alias ?alias . FILTER regex(?alias,'Dana') . }
OPTIONAL {?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }
}
</query>

<string>
Which music albums contain the song Last Christmas?
</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri ?string WHERE {
?single rdf:type onto:Single .
?single onto:album ?uri .
?single foaf:name 'Last Christmas'@en .
OPTIONAL
{?uri rdfs:label ?string .
FILTER (lang(?string) = 'en') }
}
</query>

<string>Which books were written by Danielle Steel?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT ?uri ?string WHERE {
  ?uri rdf:type onto:Book .
  ?uri onto:author ?author .
  ?author foaf:name 'Danielle Steel'@en .
OPTIONAL
  {?uri rdfs:label ?string . FILTER (lang(?string) = 'en')}
}
</query>
```

<string>Which companies are located in California, USA?</string>

```
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT DISTINCT ?uri ?string WHERE {
  ?uri rdf:type onto:Organisation .
  ?uri onto:location res:California .
OPTIONAL
  {?uri rdfs:label ?string . FILTER (lang(?string) = 'en')}
}
</query>
```

<string>Which genre does DBpedia belong to?</string>

```
<query>
PREFIX prop: <http://dbpedia.org/property/>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?uri ?string WHERE {
  ?dbpedia rdf:type onto:Software .
  ?dbpedia onto:genre ?uri .
  ?dbpedia rdfs:label 'DBpedia'@en .
OPTIONAL
  {?uri rdfs:label ?string . FILTER (lang(?string) = 'en')}
}
</query>
```

```
<query>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```

SELECT ?uri ?string WHERE {
  ?uri rdf:type onto:Country .
  ?uri onto:language ?language .
OPTIONAL
  {?uri rdfs:label ?string . FILTER (lang(?string) = 'en')}
} ORDER BY DESC(count(?language)) LIMIT 1
</query>

```

<string>Which country has the most official languages?</string>
<query>

```

PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?uri ?string WHERE {
  ?uri rdf:type onto:Country .
  ?uri onto:language ?language .
OPTIONAL
  {?uri rdfs:label ?string .
  FILTER (lang(?string) = 'en')}
}
ORDER BY DESC(count(?language)) LIMIT 1
</query>

```

<string>In which programming language is GIMP written?</string>
<query>

```

PREFIX prop: <http://dbpedia.org/property/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT ?string WHERE {
  res:GIMP prop:programmingLanguage ?string .
}
</query>

```

<string>Who produced films starring Natalie Portman?</string>
<query>

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri ?string WHERE {
  ?film rdf:type onto:Film .
  ?film onto:starring ?actors .
  ?actors foaf:name 'Natalie Portman'@en .
  ?film onto:producer ?uri .
}

```

OPTIONAL

```
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }  
}  
</query>
```

<string>Give me all movies with Tom Cruise!</string>

<query>

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX onto: <http://dbpedia.org/ontology/>  
PREFIX res: <http://dbpedia.org/resource/>  
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX prop: <http://dbpedia.org/property/>  
SELECT DISTINCT ?uri ?string WHERE {  
  ?uri rdf:type onto:Film.  
  { ?uri prop:starring res:Tom_Cruise . }  
  UNION { ?uri onto:starring res:Tom_Cruise . }  
  OPTIONAL {?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }  
}  
</query>
```

<string>

In which films did Julia Roberts as well as Richard Gere play?

</string>

<query>

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX onto: <http://dbpedia.org/ontology/>  
PREFIX res: <http://dbpedia.org/resource/>  
SELECT ?uri ?string WHERE {  
  ?uri rdf:type onto:Film .  
  ?uri onto:starring res:Julia_Roberts .  
  ?uri onto:starring res:Richard_Gere.  
  OPTIONAL  
  {?uri rdfs:label ?string . FILTER (lang(?string) = 'en') }  
}  
</query>
```

<string>Give me all female German chancellors!</string>

<query>

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
PREFIX yago: <http://dbpedia.org/class/yago/>  
PREFIX prop: <http://dbpedia.org/property/>
```



```

SELECT ?uri ?string WHERE {
  ?uri rdf:type yago:FemaleHeadsOfGovernment.
  ?uri prop:office ?office .
  FILTER regex(?office, 'Chancellor of Germany').
  OPTIONAL
  {?uri rdfs:label ?string . FILTER (lang(?string) = 'en')}
}
</query>

```

<string>Give me all female German chancellors!</string>

```

<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?uri ?string WHERE {
  ?uri rdf:type yago:FemaleHeadsOfGovernment.
  ?uri prop:office ?office .
  FILTER regex(?office, 'Chancellor of Germany').
  OPTIONAL {?uri rdfs:label ?string .
  FILTER (lang(?string) = 'en')}
}
</query>

```

<string>Who wrote the book The pillars of the Earth?</string>

```

<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT ?uri ?string WHERE {
  ?books rdf:type onto:Book .
  ?books onto:author ?uri .
  ?books rdfs:label 'The Pillars of the Earth'@en .
  OPTIONAL
  {?uri rdfs:label ?string . FILTER (lang(?string) = 'en')}
}
</query>

```

<string>How many films did Leonardo DiCaprio star in?</string>

```

<query>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

```

```
PREFIX onto: <http://dbpedia.org/ontology/>
SELECT COUNT(?film) WHERE {
?film rdf:type onto:Film .
?film onto:starring ?actors .
?actors foaf:name 'Leonardo DiCaprio'@en .
}
</query>
```

```
<string>Give me all soccer clubs in the Premier League.</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX resource: <http://dbpedia.org/resource/>
SELECT DISTINCT ?uri ?string WHERE {
?uri onto:league resource:Premier_League .
OPTIONAL
{?uri rdfs:label ?string . FILTER (lang(?string) = 'en')}
}
</query>
```

```
<string>When was Capcom founded?</string>
<query>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?date WHERE {
res:Capcom prop:foundation ?date .
}
</query>
```

```
<string>When was Capcom founded?</string>
<query>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?date WHERE {
res:Capcom prop:foundation ?date .
}
</query>
```

```
<string> What is the highest mountain?</string>
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX onto: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?uri ?string WHERE {
  ?uri rdf:type onto:Mountain .
  ?uri onto:elevation ?elevation .
  OPTIONAL
  {?uri rdfs:label ?string .
   FILTER (lang(?string) = 'en')} }
ORDER BY DESC(?elevation) LIMIT 1
</query>
```

ie Portman an actress?</string>

```
<query>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX onto: <http://dbpedia.org/ontology/> ASK WHERE {
  ?subject rdf:type onto:Actor.
  ?subject rdfs:label 'Natalie Portman'@en.
}
</query>
```

Appendix B

DBR Semantic Relatedness Gold Standard

The tables of this Appendix describes the set of term pairs and the human annotations (S.1 - S.16) with regard to their degree of semantic relatedness. The column *Rel. (Mean)* describes the mean between all semantic relatedness values from human annotators. The term pairs describe entities of DBpedia (first column) and their associated query terms (second column).

Term 1	Term 2	S. 1	S. 2	S. 3	S. 4	S. 5	S. 6	S. 7	S. 8	S. 9	S. 10	S. 11	S. 12	S. 13	S. 14	S. 15	S. 16	Rel. (Mean)	Std Dev.	Var.
advisor	professor	3.5	4	3	3	4	3.5	3	3	2.5	3	2.5	2	2	3	2	2	2.88	0.67	0.45
affiliation	employment	4	2.5	3	4	3	3.5	3	3	3	0	3	4	2	2	4	3	2.96	1.00	1.00
age	old	4	4	3	3	4	3	3.4	2.5	3	2	2.5	4	3	3	2	4	3.19	0.71	0.50
alumni	studied	4	4	2	4	4	3	3.8	3	3	3.5	2	4	3	3	3	4	3.32	0.70	0.49
anthem	country	4	4	2	4	4	2	3	3	4	4	3	3	2	4	2	3	3.13	0.79	0.62
architect	building	4	4	3	2	4	3	3	3	2.5	4	3	4	3	4	1	3	3.16	0.85	0.72
architect	house	4	4	2	4	4	3	3	3	2.5	4	2	4	4	2	2	3	3.06	0.83	0.70
asylum	madhouse	4	4	0	4	1	1	4	4	4	4	3	4	1.3	0	0	4	2.58	1.81	3.27
beatified	saint	4	4	3	4	4	2.5	4	2.5	4	4	3	3	3.5	4	4	3	3.53	0.59	0.35
bird	cock	4	4	2	3	3	3	0	3	3	3.5	1	3	3	3	2	3	2.72	1.03	1.07
bird	crane	4	3	3.5	3	3	0	3.8	0	4	4	3	4	2.2	3	0	3	2.72	1.44	2.08
birth	age	4	4	3	3	4	3	3.8	2.5	4	4	3	4	3	3	2	2	3.27	0.72	0.52
birthdate	age	4	4	3	3.5	4	3	3.5	3	4	3	2.5	4	3.3	3	3	2	3.30	0.60	0.36
book	novel	4	4	3	3	4	2	3.8	2	4	4	3.5	4	3	4	4	4	3.52	0.71	0.51
book	pages	4	4	2.5	2.5	4	3	3	3	3	3	3	4	3.1	4	2.5	3	3.23	0.57	0.33
boy	lad	4	4	3.5	4	4	4	4	3.5	4	3	3.5	4	3.2	2	3.5	3	3.58	0.56	0.32
broadcast	TV	4	4	2.5	2	4	3	3.5	3.8	3	4	3	3	3.2	3	3	3	3.25	0.59	0.35
brother	monk	4	4	0	3	0	0.5	3.6	3	3	0	0	2	0	3	4	3	2.07	1.67	2.78
car	automobile	4	4	3.5	4	4	4	4	4	4	4	3	4	3.5	4	4	3	3.81	0.36	0.13
casualties	conflict	3.5	3	0	2	4	3	3.5	3	2.5	4	2.5	4	2.8	0	2	2	2.61	1.23	1.51
cemetery	woodland	1	0	0	2	3	2	0	0	0.5	1	1	0	1.5	0	0	1	0.81	0.93	0.86
ceo	president	4	4	2	4	4	3	3	3	3	3	2	3	3	3	2	3	3.06	0.68	0.46
channel	broadcast	4	4	2	2	4	2	3	3.7	3	4	3	3	2.4	3	3	3	3.07	0.72	0.52
channel	television	4	4	3	2	4	2	3	3.5	3	4	2	3	2.9	3	3	3	3.09	0.69	0.48
child	family	4	4	2.5	3	4	2.5	3	3	3	4	3	4	2.7	3	2	3	3.17	0.64	0.41
chord	smile	0	0	0	0	0	0	0	1.5	0	0	2	0	0.4	0	0	2	0.37	0.74	0.55
city	Moscow	3.5	4	3	3	4	3	4	2	4	4	3	4	3.5	4	2	3	3.38	0.70	0.48
coast	forest	3	1	0	3	0	2	2	2	1	0	2.5	2	2.3	1	0	0	1.36	1.12	1.26
coast	hill	3	2.5	0	3	3	2.5	1	3	1	0	2	2	2	2	1	2	1.88	1.01	1.02
coast	shore	4	4	3.5	3	4	4	4	4	3	4	3	4	2.2	4	4	3	3.61	0.58	0.33
color	white	4	4	3	3	3	3	4	3.7	4	4	3	4	3.4	4	3	3	3.51	0.49	0.24
combatant	conflict	4	4	2.5	2	4	3	3	3	3	3	2	4	3.6	4	2	3	3.13	0.75	0.56
commander	battle	3.5	4	2	2.5	4	2	3.5	3	2.5	3	3	4	2.9	4	3	3	3.12	0.67	0.45
conflict	battle	4	4	3	3	4	2.5	4	3.5	3	3	3	3	3.3	4	3	3	3.33	0.51	0.26
cord	smile	0	0	0	0	0	0	0	1.5	0	0	0	0	0	0	0	0	0.09	0.38	0.14
crane	implement	1	3	0	0	0	1.5	0	0	0	0	0	1	1.8	0	0	2	0.64	0.96	0.92

Term 1	Term 2	S. 1	S. 2	S. 3	S. 4	S. 5	S. 6	S. 7	S. 8	S. 9	S. 10	S. 11	S. 12	S. 13	S. 14	S. 15	S. 16	Rel. (Mean)	Std Dev.	Var.
crew	space mission	4	3	2	2	3	1	3.4	3.5	2.5	3	2.5	2	2	4	2.5	2	2.65	0.82	0.68
cup	article	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0.19	0.40	0.16
cup	entity	0	1	0	0	0	0	3	0	1.5	0	1	0	0.2	1	2	0	0.61	0.91	0.83
cup	substance	2.5	3.5	0	2	0	2.5	2	0	0	0	0	2	0.9	1	0	0	1.03	1.20	1.45
delay	racism	0	0	0	0	0	0	0	0	0	0	0	0	1.8	0	0	1	0.18	0.50	0.25
democratic	party	4	4	3	2	4	2	3.5	3	3	4	2	4	3.1	4	3	3	3.23	0.75	0.56
depiction	image	4	4	3	4	3	3	4	4	3	0	2	4	3.2	3	4	4	3.26	1.06	1.13
depiction	picture	4	3	3	4	4	4	4	4	3	0	3	4	3.4	3	4	3	3.34	1.01	1.02
depiction	picture	4	4	3	4	4	4	4	4	3	0	2	4	3.3	3	4	3	3.33	1.08	1.16
direction	combination	0	1	0	0	0	0.5	1	1	0	0	0	1	1	0	0	2	0.47	0.62	0.38
director	film	4	3	3	2	4	4	3.8	2	4	4	3	4	2.5	4	3	2	3.21	0.78	0.61
drink	ear	0	0	0	0	0	0	0	1	0	0	0	1	0.2	0	0	0	0.14	0.34	0.12
Dublin	Ireland	4	4	3	3	4	3	3.8	3.8	3	4	3.5	4	3.5	4	2	3	3.48	0.59	0.34
education	engineering	3	3	2	2	0	2	3.5	3	2.5	2	2	4	1.9	2	1	1	2.18	1.00	1.00
employee	ceo	4	4	3	2	3	3	3.5	3	3	1	3	4	2.9	4	2	3	3.03	0.83	0.68
energy	secretary	0	3	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0.38	0.81	0.65
engineer	building	3.5	2	2	2	2	3	3.5	2	3	3.5	3	4	3	4	1	2	2.72	0.88	0.77
engineer	house	3.5	3	0	1	0	0.5	3	1	3	1	1	3	2.3	1	2	2	1.64	1.17	1.37
ethnicity	race	4	3	3	3	4	2	3.8	3.5	4	4	3.5	2	3.7	4	2	3	3.28	0.75	0.56
ethnicity	religion	3	0	2	3	0	2	2	3.5	3	4	2	1	1.9	3	4	2	2.28	1.21	1.47
event	date	3.5	4	2	2	4	3	3.5	3	2.5	3	3	4	2	3	1	3	2.94	0.81	0.66
experience	work	3.5	4	2	2	4	3	3	3	2.5	3	2	4	2.2	3	2	2	2.83	0.76	0.58
fees	School	3	4	2.5	1	3	2	1.5	2	2.5	2	2	4	1.1	3	3	2	2.41	0.89	0.79
floor area	ground	4	4	2.5	3	4	3	3	4	2.5	3.5	3.5	4	2.8	2	3	4	3.30	0.66	0.44
food	fruit	3	4	3	3	3	2.5	3.8	3.5	3	3	3	4	3.4	4	2	3	3.20	0.56	0.31
food	rooster	2.5	4	3	2	3	1.5	2	2	3	3	1	3	2.2	2	2.5	2	2.36	0.72	0.52
forest	graveyard	3	0	2	2	0	1.5	0	0	1	0	1	1	0.5	1	0	1	0.88	0.90	0.82
forest	graveyard	3	0	1	2	0	2.5	0	0	1	0	1	0	1.1	0	0	1	0.79	0.98	0.97
friend	knows	3.5	2	2.5	3	0	3	3.8	3	3	3	0	3	2.1	3	1	2	2.37	1.14	1.30
furnace	stove	4	3	2.5	4	3	4	4	3.8	3	1.5	1	4	2.9	4	4	2	3.17	0.99	0.98
garrison	military	4	0	2.5	3	3	3	3.8	2	3	3	2	4	1	3	3	3	2.71	1.05	1.10
gem	jewel	4	4	0	3	4	4	4	3.8	3	4	3	4	3.1	4	4	4	3.49	1.02	1.05
genre	artist	4	3	2	2	4	1	3.5	3	2.5	2	2	4	2.4	3	2	3	2.71	0.88	0.77
genre	tango	4	2.5	0	3	4	2	3.8	1	1	2	2	3	2.2	3	1	1	2.22	1.20	1.45

Term 1	Term 2	S. 1	S. 2	S. 3	S. 4	S. 5	S. 6	S. 7	S. 8	S. 9	S. 10	S. 11	S. 12	S. 13	S. 14	S. 15	S. 16	Rel. (Mean)	Std Dev.	Var.
glass	magician	0	2	0	0	0	0	0	1	0	0	0	0	0.2	0	0	1	0.26	0.57	0.33
glass	magician	0	3	0	0	0	0	0	1	0	0	0	0	0.1	0	0	1	0.32	0.79	0.63
headquarters	company	4	4	2.5	2	4	3	3.5	3	3	3.5	3	3	1	4	3	3	3.09	0.80	0.64
industry	company	3.5	4	3	3	4	3	3.8	3	2.5	4	3	4	3	4	2	3	3.30	0.62	0.38
influenced	inspired	3	4	2.5	4	4	4	3	2.7	4	3.5	3	4	3.2	4	3.5	4	3.53	0.55	0.30
instrument	piano	4	4	2.5	3	4	3	4	3	3	3	3	4	3.3	3	3.5	3	3.33	0.51	0.26
interest	like	4	4	2.5	3	4	4	3.5	3	2.5	4	3	4	2.8	3	2	4	3.33	0.68	0.47
interest	love	3	4	2.5	3	3	3	3	2	0	0	2.5	3	2.1	2	3	3	2.44	1.08	1.16
journey	car	3.5	4	2	2	4	3	3	1.5	2.5	4	3	4	2.4	2	3	2	2.87	0.85	0.72
journey	voyage	4	4	3.5	4	4	3	4	3.8	3	3	3.5	4	3	4	4	4	3.68	0.44	0.19
killed	casualties	4	4	0	3	4	2	3.8	3.6	2.5	4	3	4	3	4	2.5	3	3.15	1.07	1.14
king	cabbage	0	0	0	0	0	0	0	1	0	2	0	1	1.5	0	0	1	0.41	0.66	0.44
known for	recognized	4	2	2.5	4	4	4	4	3.5	4	2.5	1	2	3.4	3	2	4	3.12	0.99	0.98
lad	brother	3	4	2	3	4	3	3	2	2.5	0	3	4	1.8	4	2	3	2.77	1.06	1.12
lad	wizard	0	0	0	1	0	0	3	2	1.5	1	0	0	0.3	0	0	1	0.61	0.91	0.82
language	region	3	4	2	2	0	2	2	1	3	2	2	4	2.5	4	1	4	2.41	1.20	1.44
latitude	place	4	4	2.5	2	3	3	2.5	2	3	1	2.5	3	2.5	2	2	3	2.63	0.76	0.58
leader	prime minister	3.5	4	2.5	3	3	3	3	3	4	3	3	4	3.6	3	3	3	3.23	0.45	0.20
located	city	3	4	2.5	3	3	2	3	3	4	2	2	3	3.1	3	2	2	2.79	0.66	0.44
location	country	3.5	4	2.5	3	4	3	3.5	2	3	3	0.5	4	3	2	2	4	2.94	0.96	0.93
magician	wizard	4	4	0	4	4	1	3.6	4	3	3.5	3	4	3.5	4	3	4	3.29	1.17	1.37
member	family	4	4	2	3	4	2.5	3.4	3.5	3	3	3	3	2.1	4	3	2	3.09	0.70	0.49
midday	noon	4	4	3.5	4	4	4	4	3.5	4	4	3	4	3.8	4	3.5	4	3.83	0.30	0.09
monk	oracle	3	0	0	1	3	1	2	1	2.5	2	2	2	2.5	1	0	2	1.56	1.01	1.03
monk	slave	0	0	0	1	0	0	0	1	0	0	2	1	0.2	1	0	1	0.45	0.62	0.39
monk	slave	0	0	0	0	0	0	0	1	0	0	2	0	0.2	2	0	1	0.39	0.71	0.51
month	hotel	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0.31	0.70	0.50
nationality	citizenship	4	4	3	4	4	3.5	4	4	3	3.5	3	4	3.5	4	2	3	3.53	0.59	0.35
noon	string	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00	0.00	0.00
noon	string	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00	0.00	0.00
occupation	lawyer	4	3	2.5	4	4	2	3.8	3	3	2	3	4	3	3	2	3	2.99	0.69	0.48
planet	temperature	1	4	2	1	4	1	1.2	2	3	0	2.5	3	2.1	2	1	2	1.99	1.12	1.26
population	Brazil	0	2	3	2	0	3	2.5	1	1.5	2.5	0.5	3	3	2	2	3	1.94	1.06	1.13
possibility	girl	0	0	0	2	0	0	0	0	0	1	0	1	0.5	1	0	1	0.41	0.61	0.37

Term 1	Term 2	S. 1	S. 2	S. 3	S. 4	S. 5	S. 6	S. 7	S. 8	S. 9	S. 10	S. 11	S. 12	S. 13	S. 14	S. 15	S. 16	Rel. (Mean)	Std Dev.	Var.
precedent	group	0	0	0	0	0	0	1	1	0	0	0	0	0.8	0	0	0	0.30	0.59	0.35
president	country	3	4	3	2	4	2.5	3	3.5	2.5	3	3	4	3	3	2	3	3.03	0.62	0.38
president	United States	4	4	2.5	3	2	2	3	3	2.5	4	2	2	3.3	4	3	3	2.89	0.79	0.63
prince	noble	4	4	2.5	3	4	2	3	3	1.5	3.5	2	3	3	4	3	3	3.03	0.76	0.58
problem	airport	0	0	0	1	0	0	0	0	0	1	0	3	1.8	1	0	1	0.55	0.87	0.76
product	company	3.5	4	2	3	3	2	3	3	2.5	4	2	4	2.2	2	1	3	2.76	0.87	0.75
production	hike	0	0	0	0	0	0	0	0	0	0	0.5	0	0	2	0	0	0.16	0.51	0.26
profession	engineering	3	4	3	2.5	0	3	3.8	3	4	3.5	2.5	3	3	4	2	3	2.96	0.98	0.96
professor	cucumber	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00	0.00	0.00
publisher	writer	4	4	3	3	4	3	3.5	3	2.5	3	3	4	2.2	3	2	3	3.14	0.62	0.39
race	ethnic	4	4	3.5	3	4	4	3.8	3.5	3	3	2	3	3.5	3	2	3	3.27	0.64	0.42
reason	hypertension	1	0	0	1	0	1	2.5	1.5	0	0	0	0	1.2	0	0.5	0	0.54	0.75	0.56
region	river	3	2	2	2	2	1	2.5	1	1.5	0	1	3	1.8	2	2	1	1.74	0.79	0.63
revenue	company	4	4	2.5	2	2	2.5	3.5	2	3	1	2.5	4	2.9	4	2.5	3	2.84	0.89	0.79
rooster	voyage	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0.13	0.34	0.12
saint	cleric	4	4	2.5	2	3	3	3.8	2	2.5	1	3	1	2.7	0	3	3	2.53	1.12	1.25
saint	venerated	3	4	0	3	3	3	3.5	3.5	2.5	2	3	2	3	4	3	2	2.78	0.97	0.93
senator	Illinois	0	4	2	1	0	1	3.5	1	2.5	2.5	2	3	3.2	4	2	2	2.11	1.27	1.60
sign	recess	0	0	0	3	0	0.5	2	2.5	0	0	0	0	0.3	1	0	0	0.58	1.01	1.01
space mission	lunar module	3	4	0	2	2	2	3.5	3	3	4	2.5	4	3.2	2	2	3	2.70	1.04	1.07
speaker	congress	4	4	2	3	3	2	3.5	2	3	4	2	4	2.2	3	2	3	2.92	0.81	0.65
speaks	language	3	4	2.5	3	4	3	3	3.5	3	4	2	4	2.2	3	3	3	3.21	0.57	0.33
spouse	mother	4	4	2.5	2	0	2	2	3.8	2.5	0	2	3	3.3	3	3	3	3.21	0.57	0.33
star	astronomy	4	4	3	2	4	3	3	3	3	3	3	4	3.6	4	2	3	3.23	0.66	0.44
starring	film	3.5	4	3	2	4	2	3.2	3	3	3	3	3	2.9	3	3	2	2.98	0.59	0.35
starring	participating	3.5	4	2	3	4	2	3.5	3.7	3	0	2	3	2.9	0	1	2	2.48	1.28	1.64
start date	company	3	1.5	2	0	0	1	0	1	0	3	0.5	2	1.2	0	0	2	1.08	1.08	1.16
start date	beginning	4	4	2.5	4	4	2	4	3	3	4	3	4	3.6	3	4	2	3.38	0.74	0.55
stock	CID	0	3	0	1	0	1	2	3	1.5	0	0	0	1.1	0	1	2	0.98	1.07	1.15
stock	egg	2	3	0	1	0	1	2	0	1.5	0	2	1	1.4	1	1	0	1.06	0.91	0.82
stock	jaguar	1	1	0	0	0	0	0.7	0	0	0	0.5	0	0.2	0	0	0	0.28	0.41	0.17
stock	life	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	2	0.44	0.96	0.93
stock	phone	2	2	0	0	0	0	0	0	0	0	0	0	0.9	0	0	0	0.37	0.71	0.51
stone	furnace	2.5	0	1	1	0	1	1	2	3	0	1	2	0.8	1	2	0	1.14	0.93	0.86
study	university	3	4	3	2	4	3	3.5	4	3	3.5	2	4	3	3	3.5	4	3.28	0.66	0.43
sugar	approach	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.06	0.25	0.06
term	time	3	3	0	0	0	1	3	2	2.5	3	3	0	0	0	2	3	1.59	1.38	1.91
tool	implement	3	4	2	1	3	2.5	2	2.5	3	0	0	1	1.2	2	1	4	2.01	1.24	1.54
war	battle	4	4	3	3	4	4	4	4	4	4	3.5	4	3.6	4	3	3	3.69	0.44	0.19
weapon	war	4	4	3	2	4	2.5	3.5	3.7	2.5	2	2	4	2.9	4	3.5	3	3.16	0.78	0.61
Wednesday	news	1	2	0	0	0	2	1.8	0	0	0	0	3	0.6	2	1	1	0.90	0.99	0.98
wife	woman	4	4	2.5	3	4	3	3.8	3.5	4	3	3	4	2.8	4	3	3	3.41	0.55	0.30

Appendix C

Terminology Search Queries & Results

The tables in this appendix lists the set of terminology-level keyword queries and the results for Explicit Semantic Analysis (ESA) and the associated baselines (tf/idf and tf/idf + WordNet query expansion)

query	p@5 (esa) - related	p@10 (esa) - related	rr (top related)	answered (esa)	answered (tf/idf)	answered (tf/idf + qe)
airplane	1.00	0.90	1.00	y	n	n
landing	1.00	1.00	1.00	y	y	y
aircraft	1.00	1.00	1.00	y	y	y
airport	1.00	1.00	1.00	y	y	y
airport terminal	1.00	1.00	0.33	y	y	y
athlete	1.00	1.00	0.50	y	y	y
runner	0.80	0.70	0.07	y	n	n
competition	1.00	1.00	0.08	y	n	y
running	1.00	1.00	0.06	y	n	y
bass	0.60	0.80	0.25	y	n	n
guitar	0.60	0.70	0.10	y	n	n
instrument	1.00	0.90	1.00	y	y	y
string instrument	1.00	0.90	1.00	y	y	y
battle	1.00	1.00	1.00	y	y	y
war	1.00	1.00	0.50	y	n	n
beatles	N/A	N/A	N/A	N/A	N/A	N/A
band	1.00	0.80	1.00	y	y	y
rock	1.00	0.90	0.33	y	n	n
rock band	1.00	0.90	0.50	y	y	y
show	1.00	1.00	1.00	y	y	y
books	1.00	1.00	1.00	y	n	n
school books	1.00	1.00	0.70	y	y	y
subjects	1.00	1.00	1.00	y	y	y
book collection	1.00	1.00	1.00	y	y	y
camera	0.40	0.70	0.58	y	n	n
photographic camera	0.40	0.70	0.58	y	n	n
digital camera	0.40	0.40	0.55	y	n	n
photo	0.40	0.50	1.00	y	n	n
photography	1.00	1.00	1.00	y	n	n
car	0.80	0.90	0.50	y	n	y
honda	N/A	N/A	N/A	N/A	N/A	N/A
sedan	0.80	0.80	0.25	y	n	n
cartoon	0.40	0.30	0.08	y	n	n
cell	0.80	0.90	1.00	y	n	n
biology	0.40	0.40	1.00	y	n	n
microscope	N/A	N/A	N/A	N/A	N/A	N/A
children	1.00	0.90	1.00	y	n	n
kids	0.40	0.20	0.00	n	n	n
children playing	1.00	0.90	0.63	y	y	y
city	1.00	1.00	1.00	y	y	y
street	N/A	N/A	N/A	N/A	N/A	N/A
buildings	1.00	1.00	1.00	y	y	y

	p@5 (esa) - related	p@10 (esa) - related	rr (top related)	answered (esa)	answered (tf/idf)	answered (tf/idf + qe)
commercial building	1.00	0.80	1.00	y	y	y
programming language	0.60	0.80	1.00	y	y	y
source code	0.20	0.10	0.55	y	y	y
code	0.20	0.10	1.00	y	y	y
program	1.00	0.90	0.33	y	n	y
algorithm	0.40	0.20	0.05	y	n	n
energy	N/A	N/A	N/A	N/A	N/A	N/A
energy transmission	0.60	0.30	0.63	y	y	y
energy towers	N/A	N/A	N/A	N/A	N/A	N/A
engine	0.60	0.80	0.50	y	y	y
car engine	1.00	0.90	0.50	y	y	y
scientist	0.60	0.30	1.00	y	y	y
chemist	0.40	0.60	0.00	n	n	n
chemistry	0.60	0.40	0.00	n	n	n
experiment	N/A	N/A	N/A	N/A	N/A	N/A
financial	1.00	0.80	1.00	y	n	n
financial data	N/A	N/A	N/A	N/A	N/A	N/A
gun	0.40	0.30	1.00	y	n	n
weapon	0.20	0.10	1.00	y	n	n
pistol	0.20	0.10	0.04	y	n	n
helicopter	1.00	0.80	0.50	y	y	y
hurricane	N/A	N/A	N/A	N/A	N/A	N/A
satellite image	0.60	0.40	0.33	y	y	y
clouds	N/A	N/A	N/A	N/A	N/A	N/A
tornado	N/A	N/A	N/A	N/A	N/A	N/A
justice	0.80	0.50	1.00	y	n	y
court	0.80	0.50	1.00	y	y	y
law	1.00	0.80	1.00	y	n	n
judge	0.60	0.60	1.00	y	y	y
king	0.80	0.50	1.00	y	n	n
emperor	0.60	0.50	0.11	y	n	n
notebook	N/A	N/A	N/A	N/A	N/A	N/A
laptop	0.00	0.00	0.00	n	n	n
computer	1.00	0.70	0.33	y	n	n
stock exchange	1.00	0.90	1.00	y	n	n
trading	N/A	N/A	N/A	R	R	R
NYSE	0.00	0.10	0.14	y	n	n
telescope	N/A	N/A	N/A	N/A	N/A	N/A
observatory	N/A	N/A	N/A	N/A	N/A	N/A
astronomical observation	N/A	N/A	N/A	N/A	N/A	N/A
machine	0.00	0.00	0.00	n	n	n

	p@5 (esa) - related	p@10 (esa) - related	rr (top related)	answered (esa)	answered (tf/idf)	answered (tf/idf + qe)
mechanism	0.20	0.20	0.20	y	n	n
time magazine	1.00	1.00	1.00	y	y	y
magazine	1.00	1.00	1.00	y	y	y
magazine cover	1.00	1.00	1.00	y	y	y
map	0.60	0.80	1.00	y	y	y
currency	1.00	0.90	1.00	y	y	y
money	0.80	0.90	0.14	y	y	y
marriage	0.80	0.60	0.33	y	n	n
fiance	0.00	0.40	0.70	y	n	n
bride	0.00	0.00	0.06	y	n	n
church	1.00	0.60	1.00	y	n	n
wife	0.20	0.10	0.33	y	n	n
husband	0.20	0.10	1.00	y	n	n
married couple	0.80	0.40	0.33	y	n	n
movie	1.00	0.80	0.50	y	n	y
theater	0.40	0.50	0.13	y	n	n
cinema	1.00	0.80	1.00	y	n	y
pills	1.00	1.00	1.00	y	n	n
medicine	1.00	0.90	0.50	y	n	n
capsules	1.00	0.50	0.50	y	n	n
newspaper	1.00	1.00	1.00	y	y	y
piano	1.00	0.60	1.00	y	n	n
pianist	0.40	0.30	0.00	y	n	n
concert	1.00	1.00	0.54	y	n	n
planet	1.00	1.00	1.00	y	y	y
sun	N/A	N/A	N/A	N/A	N/A	N/A
star	0.80	0.50	1.00	y	y	y
space	1.00	1.00	1.00	y	y	y
earth	1.00	1.00	0.08	y	y	y
speech	0.20	0.20	1.00	y	n	y
discourse	0.20	0.80	0.10	n	n	n
politician	1.00	1.00	1.00	y	y	y
american politician	1.00	1.00	1.00	y	y	y
president	1.00	1.00	1.00	y	y	y
painting	1.00	1.00	1.00	y	y	y
painter	0.60	0.30	0.50	y	n	n
artist	1.00	0.90	1.00	y	y	y
airplane	1.00	0.90	1.00	y	n	n
factory	0.40	0.50	1.00	y	n	n

	p@5 (esa) - related	p@10 (esa) - related	rr (top related)	answered (esa)	answered (tf/idf)	answered (tf/idf + qc)
airplane manufacturer	0.40	0.70	1.00	y	y	y
production line	1.00	1.00	0.33	y	y	y
university	1.00	1.00	1.00	y	y	y
college	1.00	1.00	1.00	y	y	y
campus	1.00	1.00	1.00	y	y	y
mathematics	N/A	N/A	N/A	N/A	N/A	N/A
math	N/A	N/A	N/A	N/A	N/A	N/A
equation	N/A	N/A	N/A	N/A	N/A	N/A
professor	0.67	0.67	0.50	y	n	n
physics	N/A	N/A	N/A	N/A	N/A	N/A
lecture	N/A	N/A	N/A	N/A	N/A	N/A
lecturer	0.40	0.60	0.50	y	n	n
religion	1.00	1.00	1.00	y	y	y
christianism	0.20	0.10	0.00	n	n	n
symbols	0.60	0.40	1.00	y	n	n
islamism	0.00	0.00	0.00	n	n	n
jewish	0.80	0.50	0.50	y	n	n
show	N/A	N/A	N/A	N/A	R	R
band	N/A	N/A	N/A	N/A	R	R
musician	1.00	1.00	1.00	y	n	n
satellite	1.00	1.00	0.33	y	y	y
orbit	1.00	1.00	1.00	y	y	y
school	1.00	1.00	1.00	y	y	y
school bus	0.60	0.50	0.28	y	y	y
weather	1.00	1.00	0.25	y	n	n
forecast	N/A	N/A	N/A	R	R	R
sunny	N/A	N/A	N/A	R	R	R
storm	N/A	N/A	N/A	R	R	R
rain	N/A	N/A	N/A	R	R	R
cloudy	N/A	N/A	N/A	R	R	R
baby	0.40	0.40	0.11	y	n	y
parent	1.00	1.00	1.00	y	y	y
football	1.00	1.00	1.00	y	y	y
soccer	1.00	1.00	0.50	y	y	y
soccer match	1.00	1.00	0.21	y	y	y
stadium	1.00	1.00	1.00	y	y	y
spaceship	1.00	1.00	1.00	y	n	n
spacecraft	1.00	1.00	1.00	y	y	y
statue of liberty	N/A	N/A	N/A	R	R	R
statue	0.20	0.20	0.20	y	n	n
graduation	1.00	1.00	1.00	y	n	n
temple	0.00	0.00	0.00	n	n	n

	p@5 (esa) - related	p@10 (esa) - related	rr (top related)	answered (esa)	answered (tf/idf)	answered (tf/idf + qe)
greek temple	0.00	0.10	0.15	y	n	n
tiger	0.00	0.00	0.00	n	n	n
animal	1.00	1.00	1.00	y	y	y
feline	0.50	0.50	0.50	y	n	n
clock	N/A	N/A	N/A	N/A	N/A	N/A
time	0.50	0.50	0.50	y	y	y
death	1.00	1.00	1.00	y	y	y
tombstone	0.60	0.50	0.33	y	n	n
RIP	N/A	N/A	N/A	N/A	N/A	N/A
tools	N/A	N/A	N/A	N/A	N/A	N/A
hammer	N/A	N/A	N/A	N/A	N/A	N/A
world	0.00	0.00	0.04	n	n	y
mapa mundi	N/A	N/A	N/A	N/A	N/A	N/A

Appendix D

Training Query Set

Set of complementary training queries which were added to 24 queries from the QALD 2011 training set.

Q1: Who is the wife of Barack Obama?

Q2: From which university did the wife of Barack Obama graduate?

Q3: Is Natalie Portman an actress?

Q4: Who is the architect of Barack Obama?

Q5: Is Albert Einstein from Germany?

Q6: How many employees does IBM have?

Appendix E

Relevance Metrics Associated to the Baseline Systems

The tables in this Appendix describes the relevance metrics associated to the evaluation of two of the baseline systems, PowerAqua and FREyA.

E.1 PowerAqua

id	query	precision	recall	f-measure
28	Which states of Germany are governed by the Social Democratic	1	1	1
29	In which films directed by Garry Marshall was Julia Roberts star	1	1	1
24	What is the highest mountain in Germany?	0.00441	1	0.00878
25	Give me the homepage of Forbes.	1	1	1
26	Give me all soccer clubs in Spain.	0.84635	0.98246	0.90934
27	What is the revenue of IBM?	1	1	1
20	Which European countries are a constitutional monarchy?	1	1	1
21	How many monarchical countries are there in Europe?	0	0	0
22	Which European union members adopted the Euro?	0.9375	1	0.96774
49	How tall is Claudia Schiffer?	1	1	1
46	What place is the highest place of Karakoram?	1	1	1
23	Which presidents of the United States had more than three childr	0.03125	0.25	0.05556
44	Which locations have more than two caves?	0.125	0.05556	0.07693
45	Which mountain is the highest after the Annapurna?	0	0	0
42	Which bridges are of the same type as the Manhattan Bridge?	0	0	0
43	Which river does the Brooklyn Bridge cross?	1	1	1
40	Who is the author of WikiLeaks?	1	1	1
41	Give me the designer of the Brooklyn Bridge.	1	1	1
1	Which companies are in the computer software?	0.01309	0.22222	0.02472
3	Give me the official websites of actors of the television show C	0	0	0
2	Which telecommunications organizations are located in Belgium?	0	0	0
5	What are the official languages of the Philippines?	0.66667	1	0.8
4	Give me the capitals of all U.S. states.	0	0	0
7	Where did Abraham Lincoln die?	1	1	1
6	Who is the mayor of New York City?	1	1	1
9	Which countries have more than two official languages?	0	0	0
8	When was the Battle of Gettysburg?	1	1	1
39	Which states of the united states possess native gold?	1	0.5	0.66667
47	What did Bruce Carver die from?	1	1	1
12	Which classis does the Millepede belong to?	1	1	1
14	Was Andrew Jackson involved in a war?	1	1	1
11	What is the area code of Berlin?	0	0	0
10	Is Michelle the wife of President Obama?	1	1	1
13	In which country is the Limerick Lake?	1	1	1
38	Which states border Utah?	0	0	0
15	What is the profession of Frank Herbert?	1	1	1
48	When did Germany join the EU?	0	0	0
17	Which state of the United States of America has the highest dens	0	0	0
16	Who is the owner of Universal Studios?	0.66667	1	0.8
19	What is the currency of the Czech Republic?	1	1	1
18	Give me all cities in New Jersey with more than 100000 inhabitan	0.5	0.25	0.33333
31	Which museum exhibits The Scream?	1	1	1
30	Is proinsulin a protein?	1	1	1
37	Who is the daughter of Bill Clinton married to?	0	0	0
36	Which monarchs of the United Kingdom were married to a German?	0.5	1	0.66667
35	Is Egypts largest city also its capital?	0	0	0
34	Through which countries flow the Yenisei river?	0.33333	0.5	0.4
33	Give me the creator of Goofy?	1	1	1
32	Which television shows were created by Walt Disney?	1	1	1
50	In which country does the Nile start?	0.4	1	0.57143

E.2 Freya

id	query	precision	recall	f-measure
28	Which states of Germany are governed by the SPD?	0.4	0.5	0.44444
50	In which country is the source confluence of Nile?	0	0	0
24	What is the highest mountain in Germany?	0	0	0
25	Give me the homepage of Forbes.	1	1	1
26	Give me all soccer club in Spain.	1	0.97368	0.98666
27	What is the revenue of IBM?	1	1	1
20	Which European countries are governed as a Constitutional monarc	1	1	1
21	How many European countries are governed as monarchy?	0	0	0
22	Which European Union member states adopted the Euro?	0.9375	1	0.96774
49	How tall is Claudia Schiffer?	1	1	1
46	What is the highest place of Karakoram?	1	1	1
23	Which presidents of the United States had more than three childr	0	0	0
44	Which locations have more than two caves?	0	0	0
45	Which mountain is the highest after the Annapurna?	0	0	0
42	Which bridges are of the same type as the Manhattan Bridge?	0.13	1	0.23009
43	Which river does the Brooklyn Bridge cross?	1	1	1
40	Who is the author of WikiLeaks?	1	1	1
41	Who designed Brooklyn Bridge?	0	0	0
1	Which companies are in the industry of computer software?	0.08182	0.75	0.14754
3	Give me the official websites of actors of the television show C	1	1	1
2	Which telecommunications companies are located in Belgium?	1	0.4	0.57143
5	What are the official languages of Philippines?	1	1	1
4	What are capitals of states of the United States?	0	0	0
7	Where did Abraham Lincoln die?	1	1	1
6	Who is the mayor of New York City?	0	0	0
9	Which countries have more than two official languages?	0	0	0
8	When was the Battle of Gettysburg?	1	1	1
13	In which country is the Limerick Lake?	1	1	1
47	What did Bruce Carver die from?	1	1	1
12	Which classis does the Millipede belong to?	1	1	1
29	In which films directed by Garry Marshall was Julia Roberts star	1	0.33333	0.5
14	Was U.S. president Jackson involved in a war?	1	1	1
11	What is the area code of Berlin?	1	1	1
10	Is the wife of President Obama called Michelle?	1	1	1
39	Which states of the United States possess minerals that are gold	1	0.5	0.66667
38	Which states border Utah?	0	0	0
15	What is the occupation of Frank Herbert?	1	1	1
48	When did Germany join the EU?	0	0	0
17	Which state of the United States of America has the highest dens	0	0	0
16	Who is the owner of Universal Studios?	0.66667	1	0.8
33	Who is creator of Goofy?	0.33333	1	0.5
18	Give me all cities in New Jersey with more than 100000 inhabitan	0	0	0
30	Is proinsulin a protein?	1	1	1
37	spouse of daughter of Bill Clinton	1	1	1
36	Which monarchs of the United Kingdom had a spouse from Germany?	1	1	1
31	Which museum exhibits The Scream by Munch?	1	1	1
35	Is Egypts largest city also its capital?	1	1	1
34	Through which countries does the Yenisei River flow?	1	1	1
19	What is the currency of Czech Republic?	1	1	1
32	Which television shows were created by Walt Disney?	1	1	1

Bibliography

- [1] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. *McKinsey Global Institute Technical Report*, 2011.
- [2] Mark Beyer. Gartner says solving ‘big data’ challenge involves more than just managing volumes of data. *Gartner Technical Report*, 2011. URL <http://www.gartner.com/newsroom/id/1731916>.
- [3] Thor Olavsrud. Data scientists frustrated by data variety, find hadoop limiting. *CIO.com Feature Article*, 2014.
- [4] Michael Franklin, Alon Halevy, and David Maier. From databases to dataspace: A new abstraction for information management. *SIGMOD Rec.*, 34(4):27–33, December 2005.
- [5] Michael L. Brodie and Jason T. Liu. The power and limits of relational technology in the age of information ecosystems. Keynote, On The Move Federated Conferences, Heraklion, Greece, October 25-29, 2010, 2011.
- [6] Thanh Tran, Tobias Mathäβ, and Peter Haase. Usability of keyword-driven schema-agnostic search: A comparative study of keyword search, faceted search, query completion and result completion. In *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part II*, ESWC’10, pages 349–364, Berlin, Heidelberg, 2010. Springer-Verlag.
- [7] John Musser. *Web 2.0 Principles and Best Practices*. O’Reilly Radar, 2007.
- [8] Soeren Auer and Sebastian Hellmann. The web of data: Decentralized, collaborative, interlinked and interoperable. Keynote at LREC 2012, 2012.
- [9] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971, November 1987.
- [10] Esther Kaufmann and Abraham Bernstein. How useful are natural language interfaces to the semantic web for casual end-users? In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC’07/ASWC’07, pages 281–294, Berlin, Heidelberg, 2007. Springer-Verlag.

- [11] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, January 2012.
- [12] Renxu Sun, Chai-Huat Ong, and Tat-Seng Chua. Mining dependency relations for query expansion in passage retrieval. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 382–389, 2006.
- [13] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '93*, pages 160–169, New York, NY, USA, 1993. ACM.
- [14] Holger Bast, Debapriyo Majumdar, and Ingmar Weber. Efficient interactive query expansion with complete search. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07*, pages 857–860, New York, NY, USA, 2007. ACM.
- [15] Carolyn J. Crouch and Bokyung Yang. Experiments in automatic statistical thesaurus construction. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '92*, pages 77–88, New York, NY, USA, 1992. ACM.
- [16] Hinrich Schütze and Jan O. Pedersen. A cooccurrence-based thesaurus and two applications to information retrieval. *Inf. Process. Manage.*, 33(3):307–318, May 1997.
- [17] Susan Gauch, Jianying Wang, and Satya Mahesh Rachakonda. A corpus analysis approach for automatic query expansion and its extension to multiple databases. *ACM Trans. Inf. Syst.*, 17(3):250–269, July 1999.
- [18] Jiani Hu, Weihong Deng, and Jun Guo. Improving retrieval performance by global analysis. In *ICPR (2)*, pages 703–706. IEEE Computer Society, 2006.
- [19] Laurence A. F. Park and Kotagiri Ramamohanarao. Query expansion using a collection dependent probabilistic latent semantic thesaurus. In Zhi-Hua Zhou, Hang Li, and Qiang Yang 0001, editors, *PAKDD*, volume 4426 of *Lecture Notes in Computer Science*, pages 224–235. Springer, 2007.
- [20] Ian H. Witten David Milne and David M. Nichols. A knowledge-based search engine powered by wikipedia. 2007.
- [21] Adenike M. Lam-Adesina and Gareth J. F. Jones. Applying summarization techniques for term selection in relevance feedback. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *SIGIR*, pages 1–9. ACM, 2001.
- [22] Youjin Chang, Iadh Ounis, and Minkoo Kim. Query reformulation using automatically generated query concepts from a document space. *Inf. Process. Manage.*, 42(2):453–468, 2006.

- [23] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 4–11, New York, NY, USA, 1996. ACM.
- [24] Reiner Kraft and Jason Zien. Mining anchor text for query refinement. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 666–674, New York, NY, USA, 2004. ACM.
- [25] Guogen Zhang, Wesley W. Chu, Frank Meng, and Gladys Kong. Query formulation from high-level concepts for relational databases. In *UIDIS*, pages 64–75, 1999.
- [26] R. Goldman and J. Widom. Dataguides: Enabling query formulation and optimization in semistructured databases. Technical Report 1997-50, Stanford InfoLab, 1997.
- [27] Vipul Kashyap and Amit Sheth. Semantic and schematic similarities between database objects: A context-based approach. *The VLDB Journal*, 5(4):276–304, December 1996.
- [28] Edward Curry. Data curation insights, kevin ashley interview, 2014.
- [29] Steve Harris and Andy Seaborne. Sparql 1.1 query language w3c recommendation. W3C Recommendation, 2013. URL <http://www.w3.org/TR/sparql11-query/>.
- [30] Jeffrey Xu Yu, Lu Qin, and Lijun Chang. Keyword search in relational databases: A survey. *IEEE Data Engeneering Bulletin*, pages 67–78, 2010.
- [31] Haofen Wang, Qiaoling Liu, Thomas Penin, Linyun Fu, Lei Zhang, Thanh Tran, Yong Yu, and Yue Pan. Semplore: A scalable ir approach to search the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):177–188, 2009. ISSN 1570-8268. The Web of Data.
- [32] Renaud Delbru, Stephane Campinas, and Giovanni Tummarello. Searching web data: An entity retrieval and high-performance indexing model. *Web Semantics: Science, Services and Agents on the World Wide Web*, 10(0):33–58, 2012. Web-Scale Semantic Information Processing.
- [33] Andreas Harth, Aidan Hogan, Jrgen Umbrich, and Stefan Decker. Swse: Objects before documents! In *Proceedings of the Billion Triple Semantic Web Challenge, in conjunction with 7th International Semantic Web Conference*, 2009.
- [34] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 652–659, New York, NY, USA, 2004. ACM.
- [35] Heiner Stuckenschmidt and Frank van Harmelen. Approximating terminological queries. In H.L. Larsen et al., editor, *Proceedings of the Proceedings of the 4th International Conference on Flexible Query Answering Systems (FQAS)'02*, number 2522 in Advances in Soft Computing, pages 329–343. Springer-Verlag, 2002.

- [36] Carlos A. Hurtado, Alexandra Poulouvasilis, and Peter T. Wood. Ranking approximate answers to semantic web queries. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pages 263–277, Berlin, Heidelberg, 2009. Springer-Verlag.
- [37] Christoph Kiefer, Abraham Bernstein, and Markus Stocker. The fundamentals of isparql: A virtual triple approach for similarity-based semantic web tasks. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 295–309, Berlin, Heidelberg, 2007. Springer-Verlag.
- [38] Beat Sprenger. Semantic crystal. ein end-user-interface zur unterstützung von ontologie-abfragen mit sparql, faculty of economics, university of zurich, 2006.
- [39] Mustafa Jarrar and Marios D. Dikaiakos. Mashql: A query-by-diagram topping sparql. In *Proceedings of the 2Nd International Workshop on Ontologies and Information Systems for the Semantic Web*, ONISW '08, pages 89–96, New York, NY, USA, 2008. ACM.
- [40] Alistair Russell, Paul R. Smart, Dave Braines, and Nigel R. Shadbolt. Nitelight: A graphical tool for semantic query construction. In *Semantic Web User Interaction Workshop (SWUI 2008)*, April 2008.
- [41] Philipp Cimiano. Orakel: A natural language interface to an f-logic knowledge base. In *Proc. of the 9th International Conference on Applications of Natural Languages to Information Systems (NLDB)*, volume 3136 of *Lecture Notes in Computer Science*, pages 401–406. Springer, 2004.
- [42] Vanessa Lopez, Andriy Nikolov, Marta Sabou, Victoria S. Uren, Enrico Motta, and Mathieu d'Aquin. Scaling up question-answering to linked data. In *Proc. of the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, volume 6317 of *Lecture Notes in Computer Science*, pages 193–210. Springer, 2010.
- [43] Christina Unger and Philipp Cimiano. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *Proceedings of the 16th International Conference on Natural Language Processing and Information Systems, NLDB'11*, pages 153–160, Berlin, Heidelberg, 2011. Springer-Verlag.
- [44] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12*, pages 639–648, New York, NY, USA, 2012. ACM.
- [45] Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. Freya: An interactive way of querying linked data using natural language. In *Proc. of the European Semantic Web Conference Workshops*, volume 7117 of *Lecture Notes in Computer Science*, pages 125–138. Springer, 2012.
- [46] Douglas Lenat. Artificial intelligence as commonsense knowledge, 2007. URL <http://www.leaderu.com/truth/2truth07.html>.

- [47] Henry Lieberman. Usable ai requires commonsense knowledge, 2007. URL <http://web.media.mit.edu/~lieber/Publications/Usable-AI-Commonsense.pdf>.
- [48] Doug Lenat, Mayank Prakash, and Mary Shepherd. Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Mag.*, 6(4):65–85, January 1986.
- [49] J. R. Firth. A synopsis of linguistic theory 1930-55. 1952–59:1–32, 1957.
- [50] Zellig Harris. Distributional structure. *Word 10 (23)*, pages 146–162, 1954.
- [51] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [52] Peter F. Brown, Peter V. de Souza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December 1992.
- [53] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- [54] Pat Helland. If you have too much data, then 'good enough' is good enough. *Commun. ACM*, 54(6):40–47, June 2011.
- [55] Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 149–157, New York, NY, USA, 2003. ACM.
- [56] Big data. Wikipedia article, 2014. URL http://en.wikipedia.org/wiki/Big_data.
- [57] Doug Laney. 3d data management: Controlling data volume, velocity, and variety. Technical Report, 2001.
- [58] Mike Loukides. What is data science? O'Reily Radar, 2010. URL <http://radar.oreilly.com/2010/06/what-is-data-science.html>.
- [59] Adam Jacobs. The pathologies of big data. *Commun. ACM*, 52(8):36–44, August 2009.
- [60] C.W. Choo. The knowing organization: How organizations use information to construct meaning, create knowledge and make decisions. *International Journal of Information Management*, 16(5):329 – 340, 1996.
- [61] Carole L. Palmer, Melissa H. Cragin, P. Bryan Heidorn, and Linda C. Smith. Data curation for the long tail of science: The case of environmental sciences. In *Proceedings of the 3rd International Digital Curation Conference*, 2005.

- [62] Serge Abiteboul, Rakesh Agrawal, Phil Bernstein, Mike Carey, Stefano Ceri, Bruce Croft, David DeWitt, Mike Franklin, Hector Garcia Molina, Dieter Gawlick, Jim Gray, Laura Haas, Alon Halevy, Joe Hellerstein, Yannis Ioannidis, Martin Kersten, Michael Pazzani, Mike Lesk, David Maier, Jeff Naughton, Hans Schek, Timos Sellis, Avi Silberschatz, Mike Stonebraker, Rick Snodgrass, Jeff Ullman, Gerhard Weikum, Jennifer Widom, and Stan Zdonik. The lowell database research self-assessment. *Commun. ACM*, 48(5):111–118, May 2005.
- [63] Antonio Badia and Daniel Lemire. A call to arms: Revisiting database design. *SIGMOD Rec.*, 40(3):61–69, November 2011.
- [64] John F. Roddick, Lina Al-Jadir, Leopoldo Bertossi, Marlon Dumas, Florida Estrella, Heidi Gregersen, Kathleen Hornsby, Jens Lufter, Federica Mandreoli, Tomi Männistö, Enric Mayol, and Lex Wedemeijer. Evolution and change in data management and issues and directions. *SIGMOD Rec.*, 29(1):21–25, March 2000.
- [65] Peter Pin-Shan Chen. The entity-relationship model—toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36, March 1976.
- [66] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970.
- [67] Lappoon R. Tang and Raymond J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the 12th European Conference on Machine Learning*, pages 466–477, 2001.
- [68] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284:34–43, 2001.
- [69] Semantic web stack. Wikipedia article, 2014. URL http://en.wikipedia.org/wiki/Semantic_Web_Stack.
- [70] Conrad Bock et al. Owl 2 web ontology language, 2012. URL <http://www.w3.org/TR/owl2-syntax/>.
- [71] Harold Boley et al. Rif core dialect, 2013. URL <http://www.w3.org/TR/rif-core/>.
- [72] Ian Horrocks et al. Swrl: A semantic web rule language combining owl and ruleml, 2013. URL <http://www.w3.org/Submission/SWRL/>.
- [73] Tim Berners-Lee. Linked data. Web page, 2006. URL <http://www.w3.org/DesignIssues/LinkedData.html>.
- [74] Christian Bizer et al. Linked data: The story so far. *International Journal on Semantic Web and Information Systems*, 5:1–22, 2009.
- [75] Tim Berners-Lee. Linked data design issues, 2006. URL <http://www.w3.org/DesignIssues/LinkedData.html>.

- [76] Marcelo Arenas et al. A direct mapping of relational data to rdf, 2012. URL <http://www.w3.org/TR/rdb-direct-mapping/>.
- [77] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. Evaluating question answering over linked data. *Web Semant.*, 21:3–13, August 2013.
- [78] Roi Blanco, Harry Halpin, Daniel M. Herzig, Peter Mika, Jeffrey Pound, Henry S. Thompson, and Thanh Tran. Repeatable and reliable semantic search evaluation. *Web Semant.*, 21:14–29, August 2013.
- [79] Krisztian Balog and Robert Neumayer. A test collection for entity search in dbpedia. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13*, pages 737–740, New York, NY, USA, 2013. ACM.
- [80] Inex linked data track, 2014. URL <https://inex.mmci.uni-saarland.de/tracks/lod/>.
- [81] Thomas Eiter, Giovambattista Ianni, Thomas Krennwallner, and Axel Polleres. Reasoning web. chapter Rules and Ontologies for the Semantic Web, pages 1–53. Springer-Verlag, Berlin, Heidelberg, 2008.
- [82] Wikipedia article. Entity attribute value model, 2014. URL http://en.wikipedia.org/wiki/Entity_attribute_value_model.
- [83] Valentin Dinu and Prakash M. Nadkarni. Guidelines for the effective use of entityattribute-value modeling for biomedical databases. *I. J. Medical Informatics*, 76(11-12):769–779, 2007.
- [84] Manuel García-Solaco, Fèlix Saltor, and Malú Castellanos. Object-oriented multidatabase systems. chapter Semantic Heterogeneity in Multidatabase Systems, pages 129–202. Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK, 1995.
- [85] Won Kim, Injun Choi, Sunit Gala, and Mark Scheevel. Modern database systems. chapter On Resolving Schematic Heterogeneity in Multidatabase Systems, pages 521–550. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995.
- [86] Joachim Hammer and Dennis Mcleod. An approach to resolving semantic heterogeneity in a federation of autonomous, heterogeneous database systems. *International Journal of Intelligent and Cooperative Information Systems*, 2(1):51–83, 1993.
- [87] David George. Understanding structural and semantic heterogeneity in the context of database schema integration, 2005.
- [88] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, September 1990.
- [89] Carlo Batini, Maurizio Lenzerini, and Shamkant B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364, December 1986.

- [90] Roberta Lamb and Elizabeth Davidson. The new computing archipelago: Intranet islands of practice, 2000.
- [91] Robert M. Colomb. Impact of semantic heterogeneity on federating databases. *The Computer Journal*, 40:235–244, 1997.
- [92] Christine Parent and Stefano Spaccapietra. Issues and approaches of database integration. *Commun. ACM*, 41(5es):166–178, May 1998.
- [93] Heiner Stuckenschmidt. Query processing on the semantic web, knstliche intelligenz. *Kuenstliche Intelligenz - KI*, 17(3), 2003.
- [94] Giovanni Tummarello, Renaud Delbru, and Eyal Oren. Sindice.com: Weaving the open linked data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07*, pages 552–565, Berlin, Heidelberg, 2007. Springer-Verlag.
- [95] Renaud Delbru, Nickolai Toupikov, Michele Catasta, and Giovanni Tummarello. A node indexing scheme for web entity retrieval. In Lora Aroyo, Grigoris Antoniou, Eero Hyvnen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *The Semantic Web: Research and Applications*, volume 6089 of *Lecture Notes in Computer Science*, pages 240–256. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13488-3.
- [96] Renaud Delbru, Nickolai Toupikov, Michele Catasta, Giovanni Tummarello, and Stefan Decker. Hierarchical link analysis for ranking web data. In *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part II, ESWC'10*, pages 225–239, Berlin, Heidelberg, 2010. Springer-Verlag.
- [97] Daniel J. Abadi, Adam Marcus, Samuel R. Madden, and Kate Hollenbach. Using the barton libraries dataset as an rdf benchmark. Technical report, MIT, 2007.
- [98] Aidan Hogan, Andreas Harth, Jürgen Umbrich, Sheila Kinsella, Axel Polleres, and Stefan Decker. Searching and browsing linked data with swse: The semantic web search engine. *Web Semant.*, 9(4):365–401, December 2011.
- [99] Aidan Hogan, Andreas Harth, and Axel Polleres. Saor: Authoritative reasoning for the web. In *In Proceedings of the 3rd Asian Semantic Web Conference (ASWC 2008), Bangkok*, 2008.
- [100] Andreas Harth, Jrgen Umbrich, Aidan Hogan, and Stefan Decker. Yars2: A federated repository for querying graph structured data from the web. In *of Lecture Notes in Computer Science*, pages 211–224. Springer, 2007.
- [101] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. An evaluation of knowledge base systems for large owl datasets. In *In International Semantic Web Conference*, pages 274–288. Springer, 2004.
- [102] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, pages 697–706, New York, NY, USA, 2007. ACM.

- [103] Xin Dong and Alon Halevy. Indexing dataspace. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 43–54, New York, NY, USA, 2007. ACM.
- [104] Qi Zhou, Chong Wang, Miao Xiong, Haofen Wang, and Yong Yu. Spark: Adapting keyword query to semantic search. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 694–707, Berlin, Heidelberg, 2007. Springer-Verlag.
- [105] Eyal Oren, Christophe Guaret, and Stefan Schlobach. Anytime query answering in rdf through evolutionary algorithms. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 98–113. Springer Berlin Heidelberg, 2008.
- [106] Martin Svihla and Ivan Jelinek. Benchmarking rdf production tools. In Roland Wagner, Norman Revell, and Gnther Pernul, editors, *Database and Expert Systems Applications*, volume 4653 of *Lecture Notes in Computer Science*, pages 700–709. Springer Berlin Heidelberg, 2007.
- [107] Ian Horrocks and Sergio Tessaris. Querying the semantic web: a formal approach. In *Proc. of the 13th Int. Semantic Web Conf. (ISWC 2002)*, number 2342 in *Lecture Notes in Computer Science*, pages 177–191. Springer-Verlag, 2002.
- [108] Krys J. Kochut and Maciej Janik. Sparqler: Extended sparql for semantic association discovery. In *Proceedings of the 4th European Conference on The Semantic Web: Research and Applications*, ESWC '07, pages 145–159, Berlin, Heidelberg, 2007. Springer-Verlag.
- [109] Boanerges Aleman-Meza, Farshad Hakimpour, I. Budak Arpinar, and Amit P. Sheth. Swe-todblp ontology of computer science publications. *Web Semant.*, 5(3):151–155, September 2007.
- [110] Esther Kaufmann, Abraham Bernstein, and Lorenz Fischer. Nlp-reduce: A naive but domain-independent natural language interface for querying ontologies. 2007.
- [111] Esther Kaufmann, Abraham Bernstein, and R. Zumstein. A natural language interface to query ontologies based on clarification dialogs. In *Proc. of the 5th International Semantic Web Conference (ISWC)*, 2006.
- [112] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [113] Abraham Bernstein, Esther Kaufmann, Christian Kaiser, and Christoph Kiefer. Ginseng a guided input natural language search engine for querying ontologies. In *In Jena User Conference*, 2006.

- [114] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: Implementing the semantic web recommendations. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, WWW Alt. '04, pages 74–83, New York, NY, USA, 2004. ACM.
- [115] John Brooke. Sus: A quick and dirty usability scale. In *Usability Evaluation in Industry*. Taylor and Francis, 1996.
- [116] Esther Kaufmann. Talking to the semantic web - query interfaces to ontologies for the casual user. In *Proc. of the 5th International Semantic Web Conference (ISWC)*, pages 980–981, 2006.
- [117] Vanessa Lopez, Miriam Fernández, Enrico Motta, and Nico Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.
- [118] Vanessa Lopez, Marta Sabou, and Enrico Motta. Powermap: Mapping the real semantic web on the fly. In *Proc. of the 5th International Semantic Web Conference (ISWC)*, volume 4273 of *Lecture Notes in Computer Science*, pages 414–427. Springer, 2006.
- [119] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. Evaluation question answering over linked data. *Journal of Web Semantics*, in press.
- [120] Andre Freitas and Christina Unger. Schema-agnostic queries (saq-2015) semantic web challenge. In *Schema-agnostic Queries (SAQ-2015) Semantic Web Challenge, 12th Extended Semantic Web Conference (ESWC)*. 2015.
- [121] Andre Freitas, Joao Carlos Pereira Da Silva, and Edward Curry. On the semantic mapping of schema-agnostic queries: A preliminary study. In *Workshop of the Natural Language Interfaces for the Web of Data (NLIWoD)*, 13th International Semantic Web Conference (ISWC), Italy, 2014.
- [122] Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. Freya: An interactive way of querying linked data using natural language. In *Proceedings of the 8th International Conference on The Semantic Web, ESWC'11*, pages 125–138, Berlin, Heidelberg, 2012. Springer-Verlag.
- [123] Philipp Cimiano, Peter Haase, Jörg Heizmann, Matthias Mantel, and Rudi Studer. Towards portable natural language interfaces to knowledge bases - the case of the orakel system. *Data Knowl. Eng.*, 65(2):325–354, May 2008.
- [124] J. McCrae, G. Aguado de Cea, P. Buitelaar, P. Cimiano, T. Declerck, A. Gomez-Perez, J. Garcia, L. Hollink, E. Montiel-Ponsoda, and D. Spohr. Interchanging lexical resources on the semantic web. *Language Resources and Evaluation*, 46(4):701–719, 2012.
- [125] Aravind Joshi and Yves Schabes. Tree-adjoining grammars. In *Handbook of Formal Languages*, pages 69–123. Springer Berlin Heidelberg, 1997.
- [126] Philipp Cimiano. Flexible semantic composition with dudes. In *Proceedings of the Eighth International Conference on Computational Semantics, IWCS-8 '09*, pages 272–276, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

- [127] Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Extracting multilingual natural-language patterns for rdf predicates. In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management, EKAW'12*, pages 87–96, Berlin, Heidelberg, 2012. Springer-Verlag.
- [128] Charles L. A. Clarke, Gordon V. Cormack, D. I. E. Kisman, and Thomas R. Lynam. Question answering by passage selection (multitext experiments for trec-9). In Ellen M. Voorhees and Donna K. Harman, editors, *TREC*, volume Special Publication 500–249. National Institute of Standards and Technology (NIST), 2000.
- [129] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 41–47. ACM Press, 2003.
- [130] David Ferrucci et al. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.
- [131] Sanda M. Harabagiu, Marius A. Paşca, and Steven I. Maorano. Experiments with open-domain textual question answering. In *Proc. COLING-2000*, 2000.
- [132] Ulf Hermjakob, Eduard H. Hovy, and Chin yew Lin. Knowledge-based question answering. In *In Proceedings of the 6th World Multiconference on Systems, Cybernetics and Informatics (SCI-2002)*, pages 772–781, 2000.
- [133] Esther Kaufmann and Abraham Bernstein. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Web Semant.*, 8(4):377–393, November 2010.
- [134] Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. From keywords to semantic queries-incremental query construction on the semantic web. *Web Semant.*, 7(3):166–176, September 2009.
- [135] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. Evaluating question answering over linked data. *Web Semantics*, 21:3–13, August 2013.
- [136] R. Navigli and P. Velardi. An analysis of ontology-based query expansion strategies. In *In Proceedings of the ECML/PKDD-2003 Workshop on Adaptive Text Extraction and Mining*, 2003.
- [137] ChengXiang Zhai and John D. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM*, pages 403–410. ACM, 2001.
- [138] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *Proceedings of SIGIR*, pages 120–127, 2001.
- [139] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2):7, 2007.

- [140] Ryen W. White, Ian Ruthven, and Joemon M. Jose. A study of factors affecting the utility of implicit relevance feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–42, New York, NY, USA, 2005. ACM Press.
- [141] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan. Optimizing web search using web click-through data. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 118–126, New York, NY, USA, 2004. ACM.
- [142] J. Arguello, J. Elisas, J. Callan, and J. Carbonell. Document representation and query expansion models for blog recommendation. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18, New York, NY, USA, 2008. AAAI Press.
- [143] Christina Unger, Andre Freitas, and Philipp Cimiano. An introduction to question answering over linked data. In Manolis Koubarakis, Giorgos Stamou, Giorgos Stoilos, Ian Horrocks, Phokion Kolaitis, Georg Lausen, and Gerhard Weikum, editors, *Reasoning Web. Reasoning on the Web in the Big Data Era*, volume 8714 of *Lecture Notes in Computer Science*, pages 100–140. Springer International Publishing, 2014.
- [144] Andre Freitas, Edward Curry, Joao Gabriel Oliveira, and Sean O’Riain. Querying heterogeneous datasets on the linked data web: Challenges, approaches, and trends. *IEEE Internet Computing*, 16(1):24–33, 2012.
- [145] Charles S. Peirce. Logic as semiotic: The theory of signs. *Philosophical Writings of Peirce*.
- [146] Ricardo R. Gudwin and Fernando A. C. Gomide. Computational semiotics : An approach for the study of intelligent systems- part i : Foundations. Technical report, Unicamp, 1997.
- [147] Daniel Chandler. Semiotics for beginners. Technical report, 2014.
- [148] Susanne K Langer. *Philosophy in a New Key: A Study in the Symbolism of Reason, Rite and Art*. 1951.
- [149] Sebastian Loebner. *Understanding Semantics*. Routledge, 2014.
- [150] Peter Bogh Andersen. *A Theory of Computer Semiotics: Semiotic Approaches to Construction and Assessment of Computer Systems*. Cambridge University Press, New York, NY, USA, 1st edition, 1996.
- [151] Jamshaid Ashraf, Richard Cyganiak, Sean O’Riain, and Maja Hadzic. Open ebusiness ontology usage: Investigating community implementation of goodrelations. In *LDOW*, 2011.
- [152] James R. Hurford. The neural basis of predicate-argument structure. *Behavioral and Brain Sciences*, 26:261–283, 2003.
- [153] Peter Pin-Shan Chen. English, chinese and er diagrams. *Data and Knowledge Engineering*, 23(1):5–16, 1997. Natural Language for Data Bases (Workshop 1996).

- [154] Tomasz Imielinski and Witold Lipski, Jr. A systematic approach to relational database theory. In *Proceedings of the 1982 ACM SIGMOD International Conference on Management of Data*, SIGMOD '82, pages 8–14, New York, NY, USA, 1982. ACM.
- [155] Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technology*, 9: 5–110, 2014.
- [156] Barbara Abbott. The formal approach to meaning: Formal semantics and its recent developments. *Journal of Foreign Languages*, pages 2–20, 1999.
- [157] Ted Briscoe. Introduction to formal semantics for natural language. Technical report, 2011. URL <https://www.cl.cam.ac.uk/teaching/1011/L107/semantics.pdf>.
- [158] Lawrence W. Barsalou. *Cognitive psychology: an overview for cognitive scientists*. Lawrence Erlbaum, 1992.
- [159] Brent Berlin and Paul Kay. *Basic Color Terms: their Universality and Evolution*. University of California Press, Berkeley and Los Angeles, 1969.
- [160] Marvin Minsky. A framework for representing knowledge. Technical report, MIT-AI Laboratory Memo 306, 1974.
- [161] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.
- [162] Ora Lassila and Deborah McGuinness. The role of frame-based representation on the semantic web. Technical report, 2001.
- [163] Dirk Geeraerts. The theoretical and descriptive development of lexical semantics. *Lexicon in Focus. Competition and Convergence in Current Lexicology*, pages 23–42, 2012.
- [164] Ferdinand de Saussure. *Course in General Linguistics (translated by Roy Harris)*. 1916.
- [165] Susanne K Langer. *Hidden Myth: Structure and Symbolism in Advertising*. 1975.
- [166] Magnus Sahlgren. The distributional hypothesis. In *Rivista di Linguistica (Italian Journal of Linguistics)*, volume 20, pages 33–53, 2008.
- [167] Alessandro Lenci. Distributional approaches in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31, 2008.
- [168] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010.
- [169] Douwe Kiela and Steve Clark. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, 2014.

- [170] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.
- [171] Thomas K Landauer and Susan T. Dutnais. A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997.
- [172] Magnus Sahlgren. An introduction to random indexing. In *In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, 2005.
- [173] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1):13–47, March 2006.
- [174] Iryna Gurevych. Using the structure of a conceptual network in computing semantic relatedness. In *Proceedings of the Second International Joint Conference on Natural Language Processing, IJCNLP’05*, pages 767–778, Berlin, Heidelberg, 2005. Springer-Verlag.
- [175] Christopher A. Brewster. *Mind the Gap: Bridging from Text to Ontological Knowledge*. PhD thesis, University of Sheffield, 2008.
- [176] Andre Freitas, Joao C. Pereira da Silva, Sean O’Riain, and Edward Curry. Distributional relational networks. In *AAAI Fall Symposium Series*, 2013.
- [177] Andre Freitas, Siegfried Handschuh, and Edward Curry. Distributional-relational models: Scalable semantics for databases. In *AAAI Spring Symposium Series*, 2014.
- [178] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.
- [179] Shannon Pollard and Alan W. Biermann. A measure of semantic complexity for natural language systems. In *Proc. of the 2000 NAACL-ANLP Workshop on Syntactic and Semantic Complexity in Natural Language Processing Systems*, pages 42–46, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
- [180] Dan Melamed. Automatic construction of clean broad-coverage translation lexicons. *CoRR*, cmp-lg/9607037, 1996.
- [181] Brigitte Bigi. Using kullback-leibler distance for text categorization. In Fabrizio Sebastiani, editor, *Advances in Information Retrieval*, volume 2633 of *Lecture Notes in Computer Science*, pages 305–319. Springer Berlin Heidelberg, 2003.
- [182] Dan Melamed. Measuring semantic entropy. In *Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics*, pages 41–46, 1997.
- [183] Andre Freitas, Juliano Efon Sales, Siegfried Handschuh, and Edward Curry. How hard is the query? measuring the semantic complexity of schema-agnostic queries. In *11th International Conference on Computational Semantics (IWCS)*, 2015.

- [184] Barbara H. Partee and Vladimir. Borschev. Sortal, relational, and functional interpretations of nouns and russian container constructions. *Journal of Semantics*, 2012.
- [185] Sebastian. Loebner. Definites. *Journal of Semantics*, (4):279–326, 1985.
- [186] Sebastian Loebner. Approaches to discourse anaphora. *Proceedings of DAARC96/Discourse Anaphora and Resolution Colloquium*, 1996.
- [187] J. Voelker. *Learning Expressive Ontologies: Volume 2 Studies on the Semantic Web*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009.
- [188] P. Cimiano, A. Maedche, S. Staab, and J. Voelker. *Ontology Learning*. International Handbooks on Information Systems. Springer, 2nd revised edition edition, 2009.
- [189] Anna Szabolcsi. Combinatory grammar and projection from the lexicon. In *Lexical Matters*, 1192.
- [190] Andre Freitas, Margaret Jones, Kartik Asooja, Christos Bellos, Steve Elliott, Stefan Stenfelt, Panagiotis Hasapis, Christos Georgousopoulos, Torsten Marquardt, Nenad Filipovic, Stefan Decker, and Ratnesh Sahay. Towards a semantic representation for multi-scale finite element biosimulation experiments. In *13th IEEE International Conference on BioInformatics and BioEngineering (BIBE)*. 2013.
- [191] Karen Spaerck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.
- [192] Maik Anderka and Benno Stein. The esa retrieval model revisited. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 670–671, New York, NY, USA, 2009. ACM.
- [193] Thomas Gottron, Maik Anderka, and Benno Stein. Insights into explicit semantic analysis. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1961–1964, New York, NY, USA, 2011. ACM.
- [194] S. K. M. Wong, Wojciech Ziarko, and Patrick C. N. Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '85, pages 18–25, New York, NY, USA, 1985. ACM.
- [195] Fiber bundle, 2014. URL http://en.wikipedia.org/wiki/Fiber_bundle.
- [196] Fibration, 2014. URL <http://en.wikipedia.org/wiki/Fibration>.
- [197] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [198] Andre Freitas, Edward Curry, Joao Gabriel Oliveira, and Sean O'Riain. A distributional structured semantic space for querying rdf graph data. *Int. J. Semantic Computing*, 5(4): 433–462, 2012.

- [199] Andre Freitas, Joao C. Pereira da Silva, Danilo S. Carvalho, Sean O’Riain, and Edward Curry. Representing texts as contextualized entity-centric linked data graphs. In *12th International Workshop on Web Semantics and Web Intelligence (WebS 2013), 24th International Conference on Database and Expert Systems Applications (DEXA)*, Prague, 2013.
- [200] Danilo S. Carvalho, Andre Freitas, and Joao C. P. da Silva. Graphia: Extracting contextual relation graphs from text. In Philipp Cimiano, Miriam Fernandez, Vanessa Lopez, Stefan Schlobach, and Johanna Voelker, editors, *ESWC 2013 Satellite Events*, volume 7955 of *Lecture Notes in Computer Science*, pages 236–241. Springer Berlin Heidelberg, 2013.
- [201] Andre Freitas, Danilo S. Carvalho, and Joao C. Pereira da Silva. Extracting linked data graphs from texts: An ontology-agnostic approach. In *3rd Workshop on Data Extraction and Object Search (DEOS), 29th British National Conference on Databases (BNCOD)*, Oxford, UK, 2013.
- [202] Andre Freitas, Danilo Carvalho, Joao Carlos Silva, Sean O’Riain, and Edward Curry. A semantic best-effort approach for extracting structured discourse graphs from wikipedia. In *1st Workshop on the Web of Linked Entities (WoLE 2012)*, pages 70–81, Boston, MA, 2012.
- [203] Andre Freitas, Joao Gabriel Oliveira, Edward Curry, and Se’Riain. A multidimensional semantic space for data model independent queries over rdf data. In *International Conference on Semantic Computing (ICSC)*, pages 344–351. IEEE Computer Society, 2011.
- [204] Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Comput. Linguist.*, 33(2):161–199, June 2007. ISSN 0891-2017.
- [205] Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.*, 36(4):673–721, December 2010.
- [206] Pentti Kanerva. Associative neural memories. chapter Sparse Distributed Memory and Related Models, pages 50–76. Oxford University Press, Inc., New York, NY, USA, 1993.
- [207] Tamara Polajnar, Nitish Aggarwal, Kartik Asooja, and Paul Buitelaar. Improving esa with document similarity. In *ECIR*, pages 582–593, 2013.
- [208] Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms, 1997.
- [209] Claudia Leacock, George A. Miller, and Martin Chodorow. Using corpus statistics and wordnet relations for sense identification. *Comput. Linguist.*, 24(1):147–165, March 1998.
- [210] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc of 10th International Conference on Research in Computational Linguistics, ROCLING97*, 1997.
- [211] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, 37(1):145–151, September 2006.

- [212] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.
- [213] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 136–145, London, UK, 2002. Springer-Verlag.
- [214] Siddharth Patwardhan. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *In: Proceedings of the EACL*, pages 1–8, 2006.
- [215] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October 1965.
- [216] George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6:1–28, 1991.
- [217] Lev Finkelstein, Evgeniy Gabilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131, January 2002.
- [218] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL-Demonstrations '04, pages 38–41, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
- [219] Danilo Carvalho, Cagatay Calli, Andre Freitas, and Edward Curry. Easyesa: A low-effort infrastructure for explicit semantic analysis (demo). In *13th International Semantic Web Conference (ISWC 2014)*, Rival del Garda, 2014. Springer.
- [220] David Jurgens and Keith Stevens. The s-space package: An open source package for word space models. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 30–35, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [221] Douglas R. Cutting and Jan Pedersen. Space optimizations for total ranking. In *Proceedings of the Symposium for Document Analysis and Information Retrieval*, SDAIR '96, 1996.
- [222] Andre Freitas, Edward Curry, and Sean O'Riain. A distributional approach for terminological semantic search on the linked data web. In Sascha Ossowski and Paola Lecca, editors, *SAC*, pages 384–391. ACM, 2012.
- [223] Andre Freitas, Sean O'Riain, and Edward Curry. A distributional semantic search infrastructure for linked dataspace. In *ESWC 2013 Satellite Events*, Lecture Notes in Computer Science, pages 214–218. Springer Berlin Heidelberg, 2013.
- [224] S. Harris and A. Seaborne. Sparql 1.1 query language: W3c recommendation, 2013.
- [225] Richard Cyganiak. A relational algebra for sparql. Technical report, HP HPL-2005-170, 2005.

- [226] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3):1–45, September 2009. ISSN 0362-5915.
- [227] Olaf Hartig, Christian Bizer, and Johann-Christoph Freytag. Executing sparql queries over the web of linked data. In *Proceedings of the 8th International Semantic Web Conference, ISWC '09*, pages 293–309, Berlin, Heidelberg, 2009. Springer-Verlag.
- [228] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13, EMNLP '00*, pages 63–70, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
- [229] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03*, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [230] Joakim Nivre. Dependency grammar and dependency parsing. Technical report, Vaxjo University, 2005.
- [231] Lucien Tesniere. *Elements de syntaxe structurale*. 1959.
- [232] Michael A. Covington. A fundamental algorithm for dependency parsing. In *In Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102, 2001.
- [233] Marie Catherine De Marneffe and Christopher D. Manning. Stanford typed dependencies manual, 2008.
- [234] Rada Mihalcea. Using wikipedia for automatic word sense disambiguation, 2007.
- [235] Andre Freitas, Joao Gabriel Oliveira, Sean O’Riain, Edward Curry, and Joao Carlos Pereira da Silva. Querying linked data using semantic relatedness: A vocabulary independent approach. In *Natural Language Processing and Information Systems*, volume 6716 of *Lecture Notes in Computer Science*, pages 40–51. Springer Berlin Heidelberg, 2011.
- [236] Andre Freitas and Edward Curry. Natural language queries over heterogeneous linked data graphs: A distributional-compositional semantics approach. In *Proceedings of the 19th International Conference on Intelligent User Interfaces, IUI '14*, pages 279–288, New York, NY, USA, 2014. ACM.
- [237] Andre Freitas, Sean O’Riain, and Edward Curry. Crossing the vocabulary gap for querying complex and heterogeneous databases: A distributional-compositional semantics perspective. In *3rd Workshop on Data Extraction and Object Search (DEOS), 29th British National Conference on Databases (BNCOD)*, Oxford, UK, 2013.

- [238] Andre Freitas, Joao Gabriel Oliveira, Sean O’Riain, Edward Curry, and Joao C. Pereira da Silva. Treo: Combining entity-search, spreading activation and semantic relatedness for querying linked data. In *1st Workshop on Question Answering over Linked Data (QALD-1)*, 2011.
- [239] Andre Freitas, Joao Gabriel Oliveira, Sean O’Riain, Edward Curry, and Joao Carlos Pereira da Silva. Treo: Best-effort natural language queries over linked data. In *Natural Language Processing and Information Systems*, volume 6716 of *Lecture Notes in Computer Science*, pages 286–289. Springer Berlin Heidelberg, 2011.
- [240] Juliano E. Sales, Andre Freitas, Siegfried Handschuh, and Brian Davis. Linse: A distributional semantics entity search engine. In *The 38th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR ’15*, 2015.
- [241] Siamak Barzagar, Juliano E. Sales, Andre Freitas, Siegfried Handschuh, and Brian Davis. Dinfra: A one stop shop for computing multilingual semantic relatedness. In *The 38th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR ’15*, 2015.
- [242] Andre Freitas and Edward Curry. Do it yourself (diy) jeopardy qa system. In *The 12th International Semantic Web Conference (ISWC2013)*, 2013.
- [243] Andre Freitas, Fabricio F. de Faria, Sean O’Riain, and Edward Curry. Answering natural language queries over linked data graphs: a distributional semantics approach. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR ’13*, pages 1107–1108, 2013.
- [244] Lappoon R. Tang and Raymond J. Mooney. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the 12th European Conference on Machine Learning (ECML-2001), Freiburg, Germany*, pages 466–477, 2001.
- [245] Bettina Berendt et al. Usewod2013 3rd international workshop on usage analysis and the web of data. In *10th ESWC Semantics and Big Data*, Montpellier, France. Springer, 2013.
- [246] Combinatory categorial grammar. Wikipedia article, 2014. URL http://en.wikipedia.org/wiki/Combinatory_categorial_grammar.
- [247] Hugo Liu and Push Singh. Conceptnet; a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, October 2004.
- [248] Robert Speer, Catherine Havasi, and Henry Lieberman. Analogyspace: Reducing the dimensionality of common sense knowledge. In *In Proc. of the 23rd Intl. Conf. on Artificial Intelligence*, pages 548–553, 2008.
- [249] Trevor Cohen, Roger W. Schvaneveldt, and Thomas .C. Rindflesch. Predication-based semantic indexing: Permutations as a means to encode predications in semantic space. In *T. AMIA Annu Symp Proc.*, pages 114–118, 2009.

- [250] Vit Novacek, Siegfried Handschuh, and Stefan Decker. Getting the meaning right: A complementary distributional layer for the web semantics. In *Proceedings of the Intl. Semantic Web Conference*, pages 504–519, 2011.
- [251] Vit Novacek, Tudor Groza, Siegfried Handschuh, and Stefan Decker. Coraal - dive into publications, bathe in the knowledge. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2–3), 2010.
- [252] Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Web Semant.*, 6(4):291–308, November 2008.
- [253] Edward Grefenstette. Towards a formal distributional semantics: Simulating logical calculi with tensors. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pages 1–10. Association for Computational Linguistics, 2013. URL <http://aclweb.org/anthology/S13-1001>.
- [254] Joao Carlos Pereira da Silva and Andre Freitas. Towards an approximative ontology-agnostic approach for logic programs. In Christoph Beierle and Carlo Meghini, editors, *Foundations of Information and Knowledge Systems*, volume 8367 of *Lecture Notes in Computer Science*, pages 415–432. Springer International Publishing, 2014.
- [255] Andre Freitas, Joao Carlos Pereira da Silva, Edward Curry, and Paul Buitelaar. A distributional semantics approach for selective reasoning on commonsense graph knowledge bases. In Elisabeth Metais, Mathieu Roche, and Maguelonne Teisseire, editors, *Natural Language Processing and Information Systems*, volume 8455 of *Lecture Notes in Computer Science*, pages 21–32. Springer International Publishing, 2014.
- [256] Andre Freitas and Joao C. Pereira da Silva. Semantics at scale: When distributional semantics meets logic programming. In *ALP Newsletter*, 2014.
- [257] Andre Freitas, Edward Curry, and Siegfried Handschuh. Towards a distributional semantic web stack. In *Proceedings of the 10th International Workshop on Uncertainty Reasoning for the Semantic Web*, pages 49–52, 2014.