UAM
UNIVERSIDAD AUTONOMA
DE MADRID

Escuela Politécnica Superior

Departamento de Ingeniería Informática

Semantically enhanced Information Retrieval: an ontology-based approach

Dissertation written by

Miriam Fernández Sánchez

Under the supervision of

Pablo Castells Azpilicueta

Madrid, January 2009

Contents

List of Figures

List of Tables

Abstract

The amount of content stored and shared on the Web and other document repositories keeps increasing steadily and fast. This growth results in well known difficulties and problems when it comes to **finding and properly managing information in massive volumes**. Striking progress has been achieved in the last decade with the development of search engine technologies, which collect, store and pre-process worldwide-scale information to return relevant resources instantly in response to users' needs. However, users still miss or need considerable effort sometimes to reach their targets, even if the sought information is present in the search space.

A common cause for this is that **currently consolidated content description and query processing techniques for Information Retrieval (IR) are based on keywords**, and therefore provide limited capabilities to grasp and exploit the conceptualizations involved in user needs and content meanings. This involves limitations such as the inability to describe relations between search terms (e.g., "hurricanes originated in Mexico" vs. "hurricanes that have affected Mexico", "books about recommender systems" vs. "systems that recommend books"), or the weakness to properly cope with linguistic phenomena such as polisemy (e.g., "mouth" as part of the body vs. "mouth" as the point where a stream issues into a larger body of water) or synonymy (e.g., find "movies" when the user queries for "films").

Aiming to solve the limitations of keyword-based models, the **idea of conceptual search, understood as searching by meanings rather than literal strings**, has been the focus of a wide body of research in the **IR** field. More recently, it has been used as a prototypical scenario (or even envisioned as a potential "killer app") in the **Semantic Web (SW)** vision since its emergence in the late nineties. However the undertakings in information search and retrieval from the semantic-based technology area have not yet taken full advantage of the technologies, background, knowledge, and accumulated experience through several decades of work in the IR field tradition.

Starting from this position, this thesis investigates the definition of ontology-based IR models, oriented to the exploitation of domain KBs to support semantic search capabilities in large document repositories, stressing on the one hand the use of full-fledged ontologies in the semantic-based perspective, and on the other the consideration of unstructured content as the final search space. In other words, the thesis explores the use of semantic information to support more expressive queries and more accurate results, while the retrieval problem is formulated in a way that is proper of the IR field, thus drawing benefit from the state of the art in this area, and enabling more realistic and applicable approaches. This vision raises fundamental problems in order to make it possible, which are the object of this thesis.

Preface

Manual for climbing mountains

by Paulo Coelho

Choose the mountain you want to climb: don’t pay attention to what other people say, such as “that one’s more beautiful” or “this one’s easier”. You’ll be spending lots of energy and enthusiasm to reach your objective, so you’re the only one responsible and you should be sure of what you’re doing.

Know how to get close to it: mountains are often seen from far off – beautiful, interesting, full of challenges. But what happens when we try to draw closer? Roads run all around them, flowers grow between you and your objective, what seemed so clear on the map is tough in real life. So try all the paths and all the tracks until eventually one day you’re standing in front of the top that you yearn to reach.

Learn from someone who has already been up there: no matter how unique you feel, there is always someone who has had the same dream before you and ended up leaving marks that can make your journey easier; places to hang the rope, trails, broken branches to make the walking easier. The climb is yours, so is the responsibility, but don’t forget that the experience of others can help a lot.

When seen up close, dangers are controllable: when you begin to climb the mountain of your dreams, pay attention to the surroundings. There are cliffs, of course. There are almost imperceptible cracks in the mountain rock. There are stones so polished by storms that they have become as slippery as ice. But if you know where you are placing each footstep, you will notice the traps and how to get around them.

The landscape changes, so enjoy it: of course, you have to have an objective in mind – to reach the top. But as you are going up, more things can be seen, and it’s no bother to stop now and again and enjoy the panorama around you. At every meter conquered, you can see a little further, so use this to discover things that you still had not noticed.

Respect your body: you can only climb a mountain if you give your body the attention it deserves. You have all the time that life grants you, as long as you walk without demanding what can’t be granted. If you go too fast you will grow tired and give up half way there. If you go too slow, night will fall and you will be lost. Enjoy the scenery, take delight in the cool spring water and the fruit that nature generously offers you, but keep on walking.

Respect your soul: don't keep repeating "I'm going to make it". Your soul already knows that, what it needs is to use the long journey to be able to grow, stretch along the horizon, touch the sky. An obsession does not help you at all to reach your objective, and even ends up taking the pleasure out of the climb. But pay attention: also, don't keep saying "it's harder than I thought", because that will make you lose your inner strength.

Be prepared to climb one kilometer more: the way up to the top of the mountain is always longer than you think. Don't fool yourself; the moment will arrive when what seemed so near is still very far. But since you were prepared to go beyond, this is not really a problem.

Be happy when you reach the top: cry, clap your hands, shout to the four winds that you did it, let the wind - the wind is always blowing up there - purify your mind, refresh your tired and sweaty feet, open your eyes, clean the dust from your heart. It feels so good, what was just a dream before, a distant vision, is now part of your life, you did it!

Make a promise: now that you have discovered a force that you were not even aware of, tell yourself that from now on you will use this force for the rest of your days. Preferably, also promise to discover another mountain, and set off on another adventure.

Tell your story: yes, tell your story! Give your example. Tell everyone that it's possible, and other people will then have the courage to face their own mountains.

Thanks to all of you

for helping me to climb this wonderful mountain called PhD

Acknowledgements

The last year has been for personal reasons the most difficult of my life until now. That is why this thesis, more than to anyone else, is dedicated to all of you who with your hugs, your love, your smiles and your unconditional support, have forced me to move forward. And here, I know I do not need to write names.

But a thesis is not only composed of the last year, and doing this PhD has allowed me to share wonderful moments with lots of people who, in one way or another, have helped me to climb this mountain.

Thanks to Pablo Castells, my supervisor, for his guidance, his effort and his work during these years. Thanks for encouraging me to grow professionally, and for allowing me to collaborate with other research groups, participate in European and national projects and to attend several conferences where I have had the chance to learn a lot from people of different nationalities and institutions.

My most sincere thank to "mis chavales", past and present members of the NETS research group for all the wonderful memories that I keep of the moments we shared together: Ivan, Alex, Fernando, Sergio, Mariano, Chema, Míguel, Laura and Alvaro. My special gratitude and affection to David and Nacho who without, the development and testing of this work would not have been possible. And to Dani, for his kindly advice and reviews in the early chapters.

Following the EPS members, impossible to leave behind are the VPU (old GTI) members and the guys of the B-402 and B-401 labs, official competitors in the Christmas decoration! Thanks for all the laughter accompanying meals and cups of coffees: Chema, Jesús, Victor, Javi, Luis, Álvaro, Marcos, Álvaro II, Jorge, Fernando, Fabri, Ana, Helena, Irene, Javi, David, and Rafa.

My special thanks to José Dorronsoro and Gonzalo Martinez, for guiding and helping me to make my first steps as a teacher. Teaching algorithms has been a rewarding experience for me.

There are a still a lot of people in EPS whom I should thank, but I specially want to mention those ones that have made the experience of climbing this mountain much more bearable: Rosa Carro, Alvaro Ortigosa, Estefanía Martín, Pablo Haya, Manuel Cebrián, Manuel Freire and Javier Tejedor. Special thanks to Juana Calle and Marisa Moreno for all their help with the administrative tasks. A warm hug to all the EPS cleaning team, that have patiently dealt with my desk full of papers, and to Gloria and Jose, for all those marvellous cups of coffee.

Beyond UAM realms, I want to extend my gratitude to the people of KMi. This research institution has been a second home for me. My special thanks to Enrico Motta, Vanessa Lopez, Marta Sabou and Victoria Uren, for showing me your incredible enthusiasm towards research. Thanks to Enrico, for all the times you have made me feel like a “super student” to Marta and Victoria, for all the valuable lessons you have teach to this “little busy bee” to Marta, thanks a lot for your incredibly valuable comments of this thesis! and, to Vane, over all things, thanks a lot for your friendship chiquita! But my experiences at KMi would not have been the same without all those fabulous people of MK. Big hugs to my awesome housemates for making me feel a part of a large international and incredibly nice family: Alex, Nourdin, Patrick, Frank, Yvonne and Farah, but special thanks to Farah because you were the main pillar for all of us! To the “octopus” team: thanks to Dinariño, Ana, “El chati”, Pejman, George and Liliana. Thanks for all those sensational expeditions and moments that we have shared together, and special thanks for my yearly Bradd Pitt Calendar!

From my stay at Google, Switzerland my warm acknowledgment to the History team, Enrique Alfonseca, María Ruiz, Kathryn Cullen, Manuel Marquez, Momchil Rusinov, José Florido, Joaquín Cuenca, Dario Bigongiari, Iulian Dragos and Alejandro Forero for all these salsa nights, football-table matches and cups of coffee. Very special thanks to Alejandro Forero for his brilliant comments and help with this document, but overall, for all the valuable lessons he has taught me about life.

From the UPM thanks to Ángel López Cima, Asunción Gómez Pérez, Oscar Corcho, Raúl Palma, Raúl García and Boris Billazón, for the nice moments we have shared together starting in that excellent 2005 Summer School.

From AIFB, my kindly thanks to Denny Vrandecic, who encourage me to develop WebCore, and to Saartje Brockmans for her interest and comments on this thesis.

Huge thanks to my friends from the university, “los tinieblas”: Ada, Sandra, Javi, Jaime, Jose, Álvaro, Fer, Tito and Toño. You introduced me to this world called computer science, and your friendship guys, is one of the most valuable things I keep from our years in the university. Special thanks to you San, for all those days and nights we spent programming together.

Huge thanks to my friends from Colmenar: Ramón, Laura, Ester, Nata, Jus, Patxi, Gema, Bea, Irene, Eva, Karol, Oscar, Bruno, Delri and Jose, for all these years together, and for our laughs around a beer or a “pineapple juice”. Special thanks to Jose, because he has always encouraged me to grow professionally, and a huge hug for you Mon, because to me, you are like my own bother.

I have no words to express my gratitude to my parents Manuel and Ana, and to my brother, Isaac, for their unconditional love, continuing support and trust in me. To my brother, who, despite all our fights, always knows how to make me laugh. To my father, who has always imparted me with his insatiable interest for learning. And to my mother, because her immense capacity for work, and the joy and love she puts into everything she does are to me the most valuable lessons in life. They were and always will be my pillars.

There is one person, without whom, this thesis would have never been started, pursued or concluded. Because this thesis is more yours than anyone else, it is dedicated to you.

To Ivan

Chapter 1

Introduction

A general overview of the thesis is provided in this chapter, focusing on the definition of the problems that motivate the work, an outline of the proposals developed to address them, and the resulting outcomes of the research. Section 1.1 presents the motivation of the research, describing the problems to be addressed and the limitations of the approaches reported in literature. Section 1.2 defines the scope of the study by stating the addressed research questions, and the central sought goal. Section 1.3 summarizes the specific aimed achievements and contribution of this research to the field, as well as the approach to reach them. Section 1.4 describes the structure of this document, and finally, section 1.5 lists the publications that resulted from the research undertaken in this thesis.

1.1 Motivation

The amount of content stored and shared on the Web and other document repositories is increasing fast and continuously. This enlargement results in well known difficulties and problems, such as **finding and properly managing all the existing amount of information**. Striking progress has been achieved in the last decade with the development of search engine technologies, which collect, store and pre-process this information to return relevant resources in response to users' needs. However, users still miss or need considerable effort sometimes to reach their targets, even if the sought information is present in the search space.

A common cause for this is that **currently consolidated content description and query processing techniques for Information Retrieval (IR) are based on keywords**, and therefore provide limited capabilities to grasp and exploit the conceptualizations involved in user needs and content meanings. This involves limitations such as the inability to describe relations between search terms (e.g., "hurricanes originated in Mexico" vs. "hurricanes that have affected Mexico", "books about recommender systems" vs. "systems that recommend books"), or the weakness to properly cope with linguistic phenomena such as polisemy (e.g., "mouth" as part of the body vs. "mouth" as the point where a stream issues into a larger body of water) or synonymy (e.g., find "movies" when the user queries for "films").

Aiming to solve the limitations of keyword-based models, the **idea of conceptual search, understood as searching by meanings rather than literal strings**, has been the focus of a wide body of research in the **IR** field (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) (Dumais, 1990) (Gonzalo, Verdejo, Chugur, & Cigarrán, 1998). Some of these approaches are based on statistical methods that study the co-occurrence of terms (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) (Dumais, 1990), and therefore do not make use of a proper semantic model. Relations between terms are extracted from term frequencies without considering potential problems such as the polisemy phenomenon. Other Information Retrieval approaches make use of linguistic algorithms (Gonzalo, Verdejo, Chugur, & Cigarrán, 1998), similar to the ones used by the human mind, but rely on thesauri and taxonomies where the level of conceptualization is often shallow and sparse, especially at the level of relations.

The idea of supporting a higher-level conceptual (computerized) understanding of contents and queries has been present in the IR field since the early eighties (Croft, 1986), if not earlier (Van Rijsbergen, 1979). More recently, it can be said to have become one of the "philosopher's stones" in the **Semantic Web (SW)** community since its emergence in the late nineties. The SW vision was brought about with the aim of helping automate tasks which require a certain level of conceptual understanding of the objects involved (e.g., information objects), or the task itself, and enabling software programs to automatically find, share and combine information and resources in consistent ways. At the core of these new technologies, ontologies (Gruber, 1993) are envisioned as key elements to represent knowledge that can be understood, used and shared among distributed applications and agents. Their potential to overcome the limitations of keyword-based search in the IR context was soon envisaged and has been explored by several researchers in the SW area. However, while there have been contributions in this direction in the last few years, most achievements so far either:

a) **Make partial use of the full expressive power of an ontology-based knowledge representation**. In this case, ontologies provide a light representation of the information space, equivalent in essence to the taxonomies and thesauri used before the Semantic Web was envisioned (Christophides, Karvounarakis, Plexousakis, & Tourtounis, 2003) (Gauch, Chaffee, & Pretschner, 2003) (Guarino, Masolo, & Vetere, 1999) (Rocha, Schwabe, & Aragão, 2004). Rather than an instrument for building knowledge bases (KBs), these lightweight ontologies provide controlled vocabularies for the classification of content, and rarely surpass several KBs in size. This approach has brought improvements over classic keyword-based search through e.g., query expansion based on class hierarchies and rules on relationships, or multifaceted searching and browsing. It is not clear though that these techniques alone really take advantage of the full potential of an ontological language, beyond those that could be reduced to conventional classification schemes.

b) **Are based on Boolean retrieval models, and therefore lack an appropriate ranking scheme needed for scaling up to massive information sources**. Some semantic search techniques that do exploit large KBs in the order of GBs or TBs have been developed which handle thousands of ontology instances, classes and relations of arbitrary complexity (Castells, Foncillas, Lara, Rico, & Alonso, 2004) (Cristani & Cuel, 2005). These techniques are closer to data retrieval (plus inference) models than to IR models, and are based on an ideal view of the information space as consisting of non-ambiguous, non-redundant, formal pieces of ontological knowledge. In this view, the information retrieval problem is reduced to a Boolean retrieval task. A knowledge item is either, a correct or an incorrect answer to a given information request, thus search results are assumed to be always 100% precise, and there is no notion of approximate answer to an information need. This model makes sense when the whole information corpus can be fully represented as an ontology-driven knowledge base. However, there are limits to the extent to which knowledge can or should be formalized in this way.

First, because of the huge amount of information currently available to information systems worldwide in the form of unstructured text and media documents, converting this volume of information into formal ontological knowledge at an affordable cost is currently an unsolved problem in general. This was identified decades ago as the well-known *knowledge acquisition bottleneck* (Feigenbaum, 1997) (Feigenbaum, 1984). Second, documents hold a value of their own, and are not equivalent to the sum of their pieces, no matter how well formalized and interlinked. The replacement of a document by a bag of information atoms inevitably implies a loss of information value: the thread of thought behind the order of the sentences in free text, the choice of the words, etc., are a valuable, relevant, and necessary part of the conveyed message. Therefore, although it is useful to break documents down into smaller information units that can be reused and reassembled to serve different purposes, it is yet often appropriate to keep the original documents in the system. Third, wherever ontology values carry free text, Boolean semantic search systems do a full-text search within the string values. In fact, if the string values hold long pieces of free text, a form of keyword-based search is taking place in practice beneath the ontology-based query model since, in a way, unstructured documents are hidden within ontology values, whereby the "perfect match" assumption starts to become arguable, and search results may start to grow in size. While this may be manageable and sufficient for small KBs, the Boolean model does not scale properly for massive document repositories where searches typically return hundreds or thousands results. Boolean search systems do not generally provide clear ranking criteria, without which the search system may become useless if the search space is too big.

The main goal of this thesis is to achieve an ontology-based IR model for the exploitation of full-fledged domain ontologies and knowledge bases, to support semantic retrieval capabilities, while still retaining the view of approximate search in document repositories. In contrast to Boolean semantic search systems, in our perspective full documents, rather than (or in addition to) specific ontology values from a KB, are sought and returned in response to user information needs. For this purpose, our search system takes advantage of both detailed instance-level knowledge available in the KB, and topic taxonomies for classification. To cope with large-scale information sources, we propose an adaptation of the classic vector-space model (Salton, 1986), suitable for an ontology-based representation, upon which a ranking algorithm is defined.

Further, this thesis also aims to explore the extension of the semantic proposed search model to open and heterogeneous environments such as the World Wide Web. Achieving an effective deployment of a semantic IR model on a decentralized, heterogeneous, dynamic and massive repository of content such as the Web is a considerable challenge. As a matter of fact, the vision of introducing ontologies as key enablers for semantically enhancing search engines in this context is still unclear and remains an open problem. While ontology-based semantic search systems have been shown to perform well for organizational semantic intranets (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) (Maedche, Staab, Stojanovic, Studer, & Sure, 2003) there have not been convincing attempts yet at applying semantic search to the Web as a whole. The advancements to date are limited and partial, and can certainly not be compared to those achieved in the IR field, neither in scalability, nor in generality. Our hypothesis is that this problem has two common causes:

a) **The inability of the current ontology-based approaches to successfully scale their models and exploit the increasing amount of online available semantic metadata**. A growing amount of ontology-based semantic markup is becoming available on the Web. Research trends in the Semantic Web community view this growing semantic body as an emerging world-scale KB, with a potential cardinal impact on the future WWW (Berners-Lee, Hendler, & Lassila, 2001), enabling a new generation of intelligent applications (D'Aquin, et al., 2008).

b) **The restriction of ontology-based search systems to deal only with specific parts of the IR process**. The difference between traditional IR technologies and current approaches from the SW field starts in fact at the level of problem formulation. Most current ontology-based search approaches do not handle the IR process as a whole, where the user expresses his requirements using a set of keywords (or free text), and the system finds (ranked) answers in a set of documents. This is mainly reflected at:

    - *The level of the query*, when systems do not fully address (and leave open) usability issues. In those systems, users are required to formulate their queries in ontology query languages, or complex user interfaces.

    - *The level of the search space*, when systems are not able to manage unstructured information items, such as common textual documents. All the information needs to be translated to formal pieces of ontological knowledge before it can be used. This is not clearly scalable to massive and heterogeneous repositories, where a huge volume of unstructured content needs to be pre-processed and translated to ontological knowledge before it can be retrieved.

In order to explore the feasibility of semantic information retrieval in massive and heterogeneous environments like the Web, and as a first step in this direction, the ontology-based retrieval model proposed in this work is extended in the following aspects:

- A combined exploitation of the SW and the WWW spaces. Namely, both relevant semantic data drawn from the SW, and information found in standard Web pages, are used to answer user queries.

- Dealing with the complete IR cycle, from the expression of queries in natural language, to the formation of a ranked set of Web documents in response. Particular requirements of this goal include:

    - Not requiring users to learn any special-purpose query language. The system shall support open ended queries in natural language.

    - Providing a flexible and scalable solution to the problem of integrating data from the SW with information from standard Web pages. In particular, the proposed solution does not require hardwiring the links between Web pages and semantic markup. On the contrary these are created dynamically, leaving both information sources decoupled.

To evaluate our ontology-based retrieval model, and compare it against traditional keyword-based approaches, we need an appropriate evaluation benchmark. Information Retrieval (IR) systems have traditionally been compared against each other using standard sets of queries and corpora. However, SW search systems still lack formal and shared evaluation datasets and benchmarks. This thesis seeks pioneer work on the **creation of reusable benchmarks for evaluating ontology-based retrieval systems**, drawing from IR methodologies, datasets, and standard resources.

1.2 Research questions

The **research problem** addressed in this work can be stated as follows:

Mainstream IR technologies are based on plain keywords, and have limited expressivity to account for semantic relationships between concepts which are often key in expressing user needs and finding the answers, or in fact do generally not handle a clear notion of concepts themselves. From a different angle, other information modeling paradigms, such as relational data models or, more recently, ontology-based models, have a much higher expressive power, but cannot be directly applied to unstructured information objects, carrying free text or multimedia content.

This thesis further expands the above research problem in to the following specific **research questions**:

- *Q1: What do we understand by semantic search?*

  Proposals from different research areas have been presented in the literature as "semantic search" approaches. An important research question of this work is to seek a clear definition of the so-called "semantic search", or equivalently, distinguish and relate the different ones that have been used.

- *Q2: Where are we standing in the progress towards semantic information retrieval?*

  In order to steer potential contributions to the state of the art we first identify the main achievements and limitations towards semantic search and retrieval from different research fields, Information Retrieval (IR) and Semantic Web (SW).

- *Q3: Can we combine achievements in semantic retrieval from different research fields and thereupon give rise to enhanced retrieval models?*

  SW and IR approaches towards semantic search present different advantages and limitations. An important research question of this thesis work is whether it is feasible to join, under a common model, the main advantages of both research areas.

- *Q4: Can semantic retrieval models be scaled to open, massive, heterogeneous environments such as the World Wide Web?*

Scalability is a pending general goal of SW technologies, hindering their competitiveness against consolidated keyword-based approaches. This thesis aims to seek further progress through this barrier by exposing the proposed semantic models to a challenging search space such as the WWW.

- *Q5: How to standardize the evaluation of semantic retrieval systems?*

  Research from semantic retrieval is still a long way from defining standard evaluation benchmarks that comprise all the required information to judge the quality of the current semantic search methods. This thesis work aims to research the development of potentially widely applicable evaluation benchmarks to test the quality of semantic search approaches.

- *Q6: How to deal with the problem of knowledge incompleteness?*

  Until, if ever, ontologies and metadata (and the SW itself) become a worldwide reality, the lack or incompleteness of available semantic knowledge is a limitation we shall likely have to live with in the mid-term. This thesis researches new techniques to retain the recall and precision of keyword-based retrieval when the semantic knowledge is not available or incomplete to cover the user information needs.

Starting from the above problem statement and research questions, the **central goal** undertaken in this work can be synthesized as:

The realization of an ontology-based retrieval model that exploits domain ontologies and knowledge bases to support semantic search in large, open and heterogeneous repositories of unstructured information.

The properties of the sought model, expressed in this goal, are expanded into the following requirements, which define the starting point for the research undertaken here. The model and methods addressing our goal shall:

- Make complete use of the full expressive power of an ontology-based knowledge representation.
- Return full documents, in addition to specific ontology values from a KB, in response to user information needs.
- Provide ontology-based retrieval and ranking algorithms which cope with large-scale information sources, such as the WWW or large intranets. In particular:
  - The system should manage massive amounts of information.
  - The system should deal with heterogeneous information sources.
  - The system should provide user-friendly ways of consultation.
- Retain the recall and precision of keyword-based search when ontology information falls short.

1.3 Contributions

Our contribution falls into three major categories:

- **Study and comparison of the different views and approximations to the notion of semantic search from the IR and SW fields, identifying fundamental limitations in the state of the art**. Despite the large amount of work on conceptual search in the IR field, semantic search has been approached as a refinement or smooth extension of traditional IR techniques rather than considering radically new paradigms, until the emergence of the SW. In this work, we study the strengths and weaknesses of the proposals towards the semantic search paradigm from both the IR and the Semantic Web fields.
- **Definition and realization of a novel semantic retrieval model**. As introduced in Section 1.1, aiming to address the identified shortcomings in semantic search approaches, this thesis proposes the exploitation of fine-grained domain ontologies and KBs to improve semantic retrieval in large repositories of unstructured information, extending the general ontology-based search capabilities towards more widely applicable IR-oriented search capabilities.
- **Investigate the feasibility of semantic retrieval in the Web environment**. As a step to a proof of concept of the feasibility of semantic retrieval within large-scale and heterogeneous environments, the proposed model is modified to address scalability, heterogeneity and usability challenges.
- **Creation of semantic retrieval evaluation benchmarks**. The standardization of experimental practice in keyword-based IR has come a long way. There is however not an equivalent body of methodologies and datasets for the evaluation of semantic retrieval models. This work aims to take a step forward, starting from traditional IR evaluation measures and datasets to provide evaluation benchmarks for ontology-based retrieval technologies.

1.4 Structure of the thesis

This thesis has been divided into three main parts. The first one gives background knowledge and a general literature survey of semantic search systems from both, the SW and IR areas. The second part contains the design, implementation and evaluation of the semantic retrieval model proposed in the thesis as well as its extensions towards the open Web environment. The third part contains research extensions on specific problems arising from the mainstream thesis research direction, including the issues of heterogeneity and knowledge incompleteness. These main parts comprise several individual chapters, as follows:

Part I. Context and related work

- **Chapter 2** provides a brief overview of the IR process. The chapter also describes the main classic IR models, as well as the most common evaluation measures and methodologies.
- **Chapter 3** provides a brief overview of the semantic-based knowledge technologies. It introduces the semantic knowledge concept as well as the advancements and problems on its representation, acquisition, annotation and evaluation.
- **Chapter 4** provides a survey of the works that have attempted to solve the problem of semantic search in both, the IR and the SW areas. We investigate the achievements and limitations of these works and present a discussion over the state of the art, which aims to motivate and introduce the model proposed in this thesis.

Part II. An ontology-based Information Retrieval model

- **Chapter 5** presents our proposed semantic retrieval model. We provide a detailed description of how introducing a level of conceptualization in classical IR models can help to improve search over traditional keyword-based approaches. We also present the generated evaluation benchmark constructed to test this model, as well as the results obtained.
- **Chapter 6** describes the extensions made to our proposal as a proof of concept to scale semantic retrieval models to large-scale and heterogeneous environments such as the Web. This chapter also presents the generation of a widely applicable Web-scale evaluation framework for semantic retrieval models based on a standard IR evaluation benchmark. The results are presented and discussed at the end of the chapter.

Part III. Coping with semantic heterogeneity and incompleteness

- **Chapter 7** describes extended research done to address the problem of information heterogeneity. In the case this proposal, a SW gateway has been implemented to face the heterogeneity problem, allowing applications to exploit all the available SW information.
- **Chapter 8** deals with the problem of knowledge incompleteness. With the aim of retaining keyword-based search recall when the available semantic information is scarce or incomplete; our proposed semantic retrieval model combines in a final ranking list the results obtained by means of our ontology-based retrieval algorithms and a traditional keyword-based search approach. The target of this chapter is to study different techniques of ranking fusion to further enhance the reliability and robustness of the combined retrieval performance.
- **Chapter 9** discusses our conclusions and contributions and points out future research lines.

Each of the above chapters starts with a paragraph describing its content and internal structure. The chapters conclude with summary sections or conclusions, in case they present experimental results. In addition to the chapters, there are three appendixes containing additional information:

Appendix A lists all the abbreviations used in the thesis.

Appendix B describes the system interface of our semantic retrieval engine.

Appendix C lists the adaptation of TREC queries done for our evaluation.

1.5 Publications

The publications that have resulted from this thesis are classified in this section by the chapters and research topics they are related to.

Chapter 5

An ontology-based Information Retrieval model

- P. Castells, M. Fernández, and D. Vallet. An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. IEEE Transactions on Knowledge and Data Engineering 19(2), Special Issue on Knowledge and Data Engineering in the Semantic Web Era, February 2007, pp. 261-272.
- D. Vallet, M. Fernández, and P. Castells. An Ontology-Based Information Retrieval Model. 2nd European Semantic Web Conference (ESWC 2005). Heraklion, Greece, May 2005. A. Gómez-Pérez andJ. Euzenat (Eds.), Springer Verlag Lecture Notes in Computer Science, Vol. 3532, ISBN: 3-540-26124-9, 2005, pp. 455-470.
- D. Vallet, M. Fernández, and P. Castells. The Quest for Information Retrieval on The Semantic Web. Upgrade 6 (6), Monograph: The Semantic Web. December 2005, pp. 19-23.
- M. Fernández, D. Vallet, and P. Castells. Automatic Annotation and Semantic Search from Protégé. Demo at the 8th International Protégé Conference. Madrid, Spain, July 2005.

These publications reflect the research carried out in the design, development and evaluation of the semantic retrieval model proposed in this thesis. This work has been done within the NETS research group at UAM in close collaboration with Pablo Castells (supervisor of this thesis) and David Vallet.

Chapter 6

Semantic retrieval on the Web

- M. Fernández, V. López, M. Sabou, V. Uren, D. Vallet, E. Motta, and P. Castells. Semantic Search meets the Web. 2nd IEEE International Conference on Semantic Computing (ICSC 2008). Santa Clara, CA, USA, August 2008.
- J. Tejedor, R. García, M. Fernández, F. J. López, F. Perdrix, J. A. Macías, R. M. Gil, M. Oliva, D. Moya, J. Colás, and P. Castells. Ontology-Based Retrieval of Human Speech. 6th International Workshop on Web Semantics (WebS 2007) at the 18th International Conference on Database and Expert Systems Applications (DEXA 2007). Regensburg, Germany, September 2007.

These publications reflect the research done towards the advancements of semantic retrieval in large-scale and heterogeneous environments such as the Web. The first publication was produced in close collaboration with the Knowledge Media Institute (KMi), who are experts in SW technologies. The last publication has been done in collaboration with the HTCLab research group of UAM, who are experts in Automatic Speech Recognition, Lleida University, experts in ontology visualization, and the SEGRE Group, a Spanish news provider.

Chapter 7

Semantic knowledge gateway

- M. Fernández, I. Cantador, and P. Castells. CORE: A Tool for Collaborative Ontology Reuse and Evaluation. 4th International Workshop on Evaluation of Ontologies for the Web (EON 2006) at the 15th International World Wide Web Conference (WWW 2006). Edinburgh, UK, May 2006.
- I. Cantador, M. Fernández, and P. Castells. A Collaborative Recommendation Framework for Ontology Evaluation and Reuse. International Workshop on Recommender Systems at the 17th European Conference on Artificial Intelligence (ECAI 2006). Riva del Garda, Italy, August 2006
- I. Cantador, M. Fernández, and P. Castells. Improving Ontology Recommendation and Reuse in WebCORE by Collaborative Assessments. Workshop on Social and Collaborative Construction of Structured Knowledge at the 16th International World Wide Web Conference (WWW 2007). Banff, Canada, May 2007
- V. López, M. Fernández, E. Motta, M. Sabou, V. Uren. Question Answering on the Real Semantic Web. Poster and demo at the 6th International Semantic Web Conference (ISWC 2007). Busan, Korea, November 2007.

In order to address the heterogeneity challenge, specific research has been done in the areas of ontology reuse and multi-ontology management. The first three publications have been done within the NETS research group of the UAM University. The last publication has been done in collaboration with the Knowledge Media Institute (KMi).

Chapter 8

Coping with knowledge incompleteness by rank fusion

- M. Fernández, D. Vallet, and P. Castells. Probabilistic Score Normalization for Rank Aggregation. 28th European Conference on Information Retrieval (ECIR 2006). London, UK, April 2006. Springer Verlag Lecture Notes in Computer Science, Vol. 3936, ISBN 3-540-33347-9, 2006, pp. 553-556.
- M. Fernández, D. Vallet, and P. Castells. Using Historical Data to Enhance Rank Aggregation. 29th Annual International ACM Conference on Research and Development on Information Retrieval (SIGIR 2006), Poster Session. Seattle, WA, August 2006.

Knowledge incompleteness is an inherent problem in the use of semantics in IR, which is addressed in this thesis by dynamic rank fusion strategies. The above publications report the research undertaken in the thesis in that area, as a means to make the model robust to domain knowledge deficiencies. This work has been done within the NETS research group of the UAM.

Related publications

This section contains some of the works published as extensions of the semantic retrieval model proposed in this thesis. These extensions include research in areas like personalization, contextualization and recommender systems. Their main goal is to improve the results obtained from the semantic retrieval model by considering extra features such as user profiles, contextual information and input or feedback from other users. This set of extensions is not explained in this document, but can be found in the following publications:

Semantic personalized retrieval

- C. Dolbear, P. Hobson, D. Vallet, M. Fernández, I. Cantador, and P. Castells. Personalized Multimedia Summaries. In Y. Kompatsiaris and P. Hobson (Eds.), Semantic Multimedia and Ontologies. Springer Verlag, ISBN 978-1-84800-075-9, March 2008, pp. 165-184.

- D. Vallet, I. Cantador, M. Fernández, and P. Castells. A Multi-Purpose Ontology-Based Approach for Personalized Content Filtering and Retrieval. 1st International Workshop on Semantic Media Adaptation and Personalization (SMAP 2006). Athens, Greece, December 2006.

- A. Evans, M. Fernández, D. Vallet, and P. Castells. Adaptive Multimedia Access: From User Needs to Semantic Personalization. IEEE International Symposium on Circuits and Systems (ISCAS 2006). Kos, Greece, May 2006.

- P. Castells, M. Fernández, D. Vallet, P. Mylonas, and Y. Avrithis. Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework. 1st IFIP WG 2.12 & WG 12.4 International Workshop on Web Semantics (SWWS 2005), November 2005. R. Meersman, Z. Tari, and P. Herrero (Eds.), Springer Verlag Lecture Notes in Computer Science, Vol. 3762, ISBN: 3-540-29739-1, 2005, pp. 977-986.

The previous publications present further research in the area of semantic retrieval personalization. They include some concept-based personalization models that aim to improve the results obtained by the semantic retrieval model proposed, adapting or re-ranking the final answers according to user-preferences.

Contextual IR personalization

- Ph. Mylonas, D. Vallet, P. Castells, M. Fernández, and Y. Avrithis. Personalized information retrieval based on context and ontological knowledge. Knowledge Engineering Review 23(1), special issue on Contexts and Ontologies, March 2008, pp. 73-100.

- D. Vallet, P. Castells, M. Fernández, P. Mylonas, and Y. Avrithis. Personalized Content Retrieval in Context Using Ontological Knowledge. IEEE Transactions on Circuits and Systems for Video Technology 17(3), special issue on the convergence of knowledge engineering, semantics and signal processing in audiovisual information retrieval, March 2007, pp. 336-346.

- D. Vallet, M. Fernández, P. Castells, P. Mylonas, and Y. Avrithis. Personalized Information Retrieval in Context. 3rd International Workshop on Modeling and Retrieval of Context

(MRC 2006) at the 21st National Conference on Artificial Intelligence (AAAI 2006). Boston, USA, July 2006.

- D. Vallet, M. Fernández, P. Castells, P. Mylonas, and Y. Avrithis. A contextual personalization approach based on ontological knowledge. International Workshop on Context and Ontologies: Theory, Practice and Applications (C&O 2006) at the 17th European Conference on Artificial Intelligence (ECAI 2006). Riva del Garda, Italy, August 2006.

The above publications report on further research on contextualization methods for semantic personalized retrieval. Ontology-based contextualization models are proposed that aim to improve the results obtained by semantic personalized search, by filtering user preferences and models according to contextual information (information obtained from the search history and the user interaction with the system), just before applying such user models to personalize search result rankings.

Recommender systems

- I. Cantador, M. Fernández, D. Vallet, P. Castells, J. Picault, and M. Ribière. A Multi-Purpose Ontology-Based Approach for Personalized Content Filtering and Retrieval. In M. Wallace, M. Angelides, and Ph. Mylonas (Eds.), Advances in Semantic Media Adaptation and Personalization. Springer Verlag Studies in Computational Intelligence, Vol. 93, ISBN 978-3-540-76359-8, February 2008, pp. 25-52.
- I. Cantador, M. Szomszor, H. Alani, M. Fernández, and P. Castells. Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations. 1st International Workshop on Collective Semantics: Collective Intelligence and the Semantic Web (CISWeb 2008), at the 5th European Semantic Web Conference (ESWC 2008). Tenerife, Spain, June 2008.

These publications report on research extensions in the area of semantic recommender systems. They propose concept-based recommendation strategies that improve or complement the capabilities of our semantic search system, recommending contents (without query), or adapting semantic query answers, according to other user's preferences and content ratings.

Part I

Context and related work

Summary

The general idea of introducing higher levels of explicit semantics in IR systems has been a long sought goal which has been approached from different perspectives. In this first part of the thesis we survey the relevant research fields, namely Information Retrieval (IR) and semantic-based knowledge technologies, for a detailed overview and analysis of the state of the art in this area, the achievements in semantic search from both fields, the limitations of present results, and open problems.

Chapter 2

Information Retrieval

This chapter provides a brief introduction to the **Information Retrieval (IR) field**. Rather than providing an in-depth revision of the field, the purpose of this chapter is to provide an overview focusing on the fundamental notions needed for later reference in the chapters where the thesis contribution is developed. The overviewed IR concepts and models are well documented in the literature and further detailed descriptions can be found elsewhere (Salton, 1986) (Baeza Yates & Ribeiro Neto, 1999). Section 2.1 motivates the IR problem. Section 2.2 discusses the complete IR process, showing its different elements and tasks. Section 2.3 describes the classical IR models. Section 2.4 presents traditional IR evaluation measures, methodologies and collections. Finally, Section 2.5 gives a brief concluding summary of the chapter.

2.1 Motivation

Libraries have traditionally been the main information repositories of historic cultures. For example, the Ancient Library of Alexandria was founded around 280 BC by Ptolomeo I Soter with the purpose of preserving the Greek civilization, surrounded in Alexandria by a very conservative Egyptian civilization. It turned out to have around 700,000 scrolls. Ptolomeo II commissioned the poet and philosopher Callimachus the task of cataloguing all books and volumes of the library. He was the first librarian of Alexandria and as a result of his work, Pinakes, the first thematic catalogue (to be known in our days) of history, was created. Other examples of big libraries are the Vatican Library created around 1500 B.C. and containing about 3,600 codices and the British Museum created around 1845 and containing about 240,000 books.

Nowadays, the amount of information available in document repositories has dramatically increased, and to a very large extent, it is stored in digital format. The World Wide Web (WWW) is probably the most prominent example, with an estimation of over 20 billion documents according to the Yahoo statistics extracted in 2005[1]. This category also includes digital libraries, company intra-

[1] http://www.ysearchblog.com/archives/000172.html

nets, etc. However, **just because the content is available it does not mean that it is useful**. Inversely, the user may not always find the information he may need. This problem arose already in the early days of computer technologies. In 1930 Vannevar Bush thought about a machine called Memex, "*a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility*". In 1950 Calvin Mooers coined the term Information Retrieval" but it was not until 1960, when Maron & Kuhns defined the problem of Information Retrieval as "*adequately identifying the information content of documentary data*". Following this idea, a lot of research has been undertaken thereafter with the aim of making the information available in digital repositories universally accessible and effectively useful.

2.2 The Information Retrieval process

The final goal of an IR system can be described as the representation, storage, organization of, and access to information items (Baeza Yates & Ribeiro Neto, 1999). This section provides a brief description of the different resources, components and tasks involved in an information retrieval system. A global, abstract view of these elements is displayed in Fig 2.1. This overview of the IR process aims to introduce the main components that are developed in our semantic retrieval model (chapters 5 and 6).

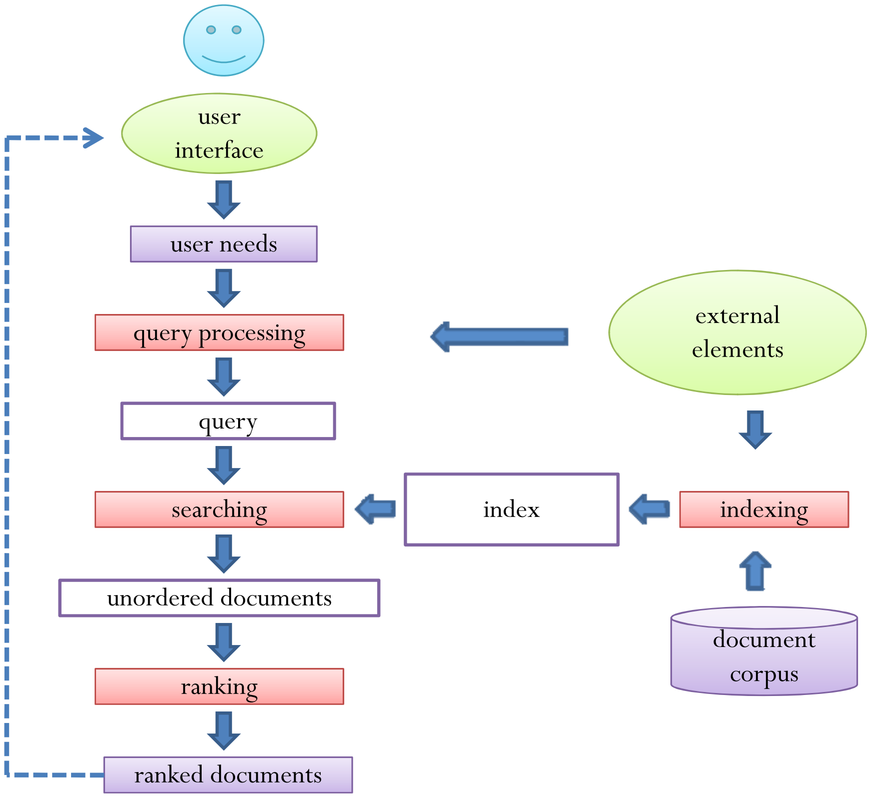


Fig 2.1 The Information Retrieval process

Input: An IR system takes two main inputs, the user needs and the information items.

- *User needs*: An information retrieval process begins when a user expresses his information need to the system. In the general case, this information need is conveyed in the form of a search string, but it can also be expressed in other formats, as in the case of Multimedia Retrieval, where the input can be an image, sound, etc.
- *Information items*: Orthogonal to the kind of queries that can be asked is the subject of the information items the system adopts. The information item is the basic element which can be retrieved as an answer to a query and it is mainly classified by its format (textual document, audio, video, image, etc) and its granularity (Web page, paragraph, sentence, etc).

Output: And IR system typically returns one main output, consisting of a ranked list of information items.

- *Ranked information items*: This output consists of a sorted list of information items. The retrieved items may have different format (text, audio, video, etc) and structure. Regarding the structure, a large classification can be made distinguishing systems that return unstructured information (items with arbitrary structure and syntax, such as free text documents), and those that return specific structured information (such as relational databases objects). The systems that return structured information are commonly characterized as data retrieval systems. While these models do cannot deal with general information about the subject or topics involved in the sought data, they return very precise answers in response to specific, unambiguous, and formally expressed information needs.

Processes: Following the work in (Croft & Harper, 1993), three main processes can be identified in an IR system: a) extraction of item content features and descriptors into a logic representation of items (*indexing*); b) handling user's information needs into an abstract representation (*query processing*) and, c) matching both representations (*searching* and *ranking*).

- *Indexing:* Not all the pieces of an information item are equally significant for representing its meaning. In written language, for example, some words carry more meaning than others. Therefore, it is usually considered worthwhile to pre-process the information items to select the elements to be used as index objects. Indices are data structures constructed to speed up search. It is worthwhile building and maintaining an index when the item collection is large and semi-static. The most common indexing structure for text retrieval is the inverted file. This structure is composed of two elements: the vocabulary and the term occurrences. The vocabulary is the set of all words in the text. For each word in the vocabulary a list of all the text positions where the word appears is stored. The set of all those lists is called occurrences.
- *Query processing:* The user needs, the query, are parsed and compiled into an internal form. In the case of textual retrieval, query terms are generally pre-processed by the same algorithms used to select the index objects. Additional query processing (e.g., query expansion) requires the use of external resources such as thesauri or taxonomies.
- *Searching:* user queries are matched against information items. As a result of this operation, a set of potential information items is returned in response to user needs. The way this is

achieved may vary considerably depending on the format of information (text, audio, video, etc), but in all cases, some form of simplification is done in the information model to make it tractable. For instance, text retrieval commonly builds on the assumption that the matching between information items (the documents) and user information needs (the query string) can be based on a set of index terms. This obviously involves an (acceptable –because reasonably effective– but considerable) loss of semantic information when text is replaced by a set of words. A similar situation occurs in multimedia retrieval where matching is performed based on numeric signal features.

- *Ranking:* The set of information items returned by the matching step generally constitutes an inexact, by nature approximate answer to the information need. Not all the items contain relevant information to the user. The ranking step aims to predict which how relevant the items are comparatively to each other, thus returning them by decreasing order of estimated relevance. Thus, in a way, ranking algorithms can be considered the core of IR systems, as they are key to determine the performance of the system.

External elements: External elements are mainly used in helping to represent, extract and process user needs and content meanings. The understanding of the semantics behind information items and users queries helps to enhance the precision of the retrieval process, and therefore, to increase user satisfaction. Three main external elements are used within IR systems: a) the *user interface*, b) *query processing operations* and c) *resources for indexing*:

- *User Interface:* A flexible user interface is needed to allow the user to express his information needs but also to express possible constraints about the information he is looking for (e.g., exact content, similar content, disjoint content, content with a specific date, language, format, etc).
- *Query processing operations:* Depending on the type of query, different mechanisms can be used to refine it. The most common ones are based on additional user input. In this spectrum, relevance feedback approaches are generally the most efficient ones. However, they reduce the usability of the systems, and therefore other external resources, such as taxonomies and thesauri, are often used instead (or complementarily) to automatically classify, disambiguate or expand query terms.
- *Resources for indexing:* Document processing resources such as thesauri and controlled vocabularies can be used to help select the terms that are more appropriate as index objects.

2.3 Modeling

As we have seen in section 2.2, the ranking algorithm is one of the main characteristic components of an IR system. A ranking algorithm operates according to basic premises regarding the notion of document relevance. Distinct sets or premises yield different IR models. The purpose of this section is to cover three of the most important classic text IR models, namely: **Boolean**, **Vector** and **Probabilistic**.

In the Boolean model document and queries are represented as a set of index terms. In the Vector-space model documents and queries are represented as vectors in a t-dimensional space. In the basic probabilistic model documents and queries representations are based on probability theory.

Following the definition in (Baeza Yates & Ribeiro Neto, 1999) an IR model is a quadruple [D,Q,*F*, sim], where:

- D is a set of (logical representations of) documents.
- Q is a set of (logical representations of) queries.
- *F* is a framework for modeling documents, queries, and their relationships.
- sim: Q × D → U is a ranking function that defines an association between queries and documents, where U is a totally ordered set (commonly [0,1], or P, or a subset thereof). This ranking and the total order in U define an order in the set of documents, for a fixed query.

To build a model, we think first of how to represent documents and user information needs. Given these representations, the framework in which they can be modeled is then conceived. This framework should also provide the intuition for constructing a ranking function. For instance, for the classic Boolean model, the framework is composed of sets of documents and the standard operations on sets. For the classic vector-space model, the framework is composed of a t-dimensional vector space and linear algebra operations on vectors. For the classic probabilistic model the framework is made of sets, standard probability operations, and the Bayes' theorem.

2.3.1 Boolean model

The Boolean Model is a simple retrieval model based on set theory and Boolean algebra. Documents are represented by the index terms extracted from documents, and queries are Boolean expressions on terms. Following the previous notation, here:

- D: the elements of D are represented as sets of index terms occurring in each document. Terms are treated as logic propositions, denoting whether the term is either present (1) or absent (0) in the document. Documents can thus be seen as the conjunction of their terms.
- Q: queries are represented as a Boolean expression composed by index terms and logic operators (AND∧, OR∨, NOT¬) which can be normalized to a disjunction of conjunctive vectors (i.e in DNF2, disjunctive normal form).
- F is a Boolean algebra over sets of terms and sets of documents.
- sim is defined by considering that a document is predicted to be relevant to a query if its index terms satisfy the query expression.

[2] http://en.wikipedia.org/wiki/Disjunctive_normal_form

Example[3]

Assume we have the query q = retrieval ∧ (text ∨ ¬multimedia).

This query is composed of three different terms: retrieval, text and multimedia, and it can be written in a disjunctive normal form as q_{dnf} = [(1,1,1) ∨ (1,1,0) ∨ (1,0,0)], where each of the components is a binary-weighted vector associated with the tuple (retrieval, text, multimedia). These binary weighted vectors are called the conjunctive components of $q_{dnf.}$

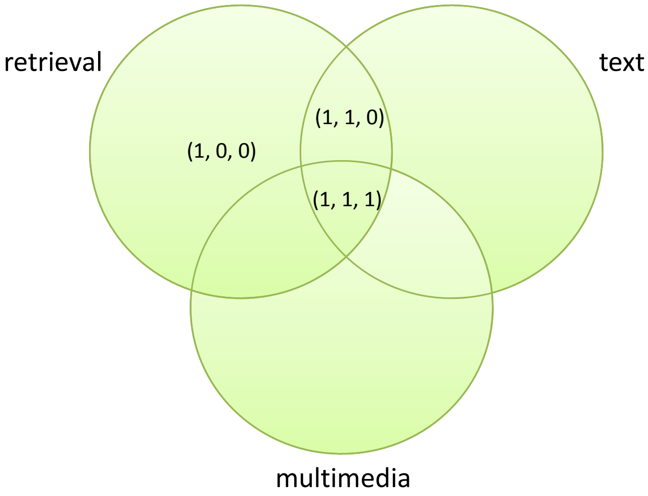


Fig 2.2 The three conjunctive components for the query q = retrieval ∧ (text ∨ ¬multimedia).

Fig 2.2 shows the set of documents containing the word retrieval, the set of documents containing the word text, and the set of documents containing the word multimedia. Given the query q, the subsets of documents that fulfill the query are: a) those containing the three terms: (1, 1, 1) b) those containing the word retrieval, but neither text nor multimedia: (1, 0, 0) and c) those containing the word retrieval and text, but not multimedia: (1, 1, 0).

Given its inherent simplicity, the Boolean model was adopted by many of the early commercial bibliographic systems. Unfortunately the Boolean model suffers from two major drawbacks. First its retrieval strategy is based on a binary criterion (i.e. a document is predicted to be either relevant or non relevant) and therefore it does not provide a proper basis for ranking the retrieved results, which may likely result in low precision levels when the retrieval space is too big. Second, it is not always easy for most users to translate an information need into a Boolean expression with logic operators, which significantly decreases the usability of the latter.

[3] Extracted from: (Baeza Yates & Ribeiro Neto, 1999)

2.3.2 Vector-space model

The vector-space model (VSM) recognizes that the use of binary weights is too limiting and proposes a framework in which partial matching is possible. This is accomplished by assigning non-binary weights to index terms in queries and documents. These terms weights are ultimately used to compute the degree of similarity between each document stored in the system and the user query. By sorting the retrieved documents in decreasing order of this degree of similarity, the VSM takes into consideration documents which match the query terms only partially. The main resulting effect is that the ranked document answer set is considerably more precise (in the sense that it better matches the user information need) than the answer set retrieved by a Boolean model.

Following the previous notation:

- D: documents are represented by a vector of words or index terms occurring in the document. Each term in the document – or, for that matter, each pair (t_i, d_j) – has a positive, non-binary associated weight $w_{i,j}$.
- Q: queries are represented as a vector of words or index terms occurring in the query. Each term in the query– or, for that matter, each pair (t_i, q) – has a positive, non-binary associated weight $w_{i,q}$.
- F is an algebraic model over vectors in a t-dimensional space.
- sim estimates the degree of similarity of a document d_j to a query q as the correlation between the vectors d_j and q. This correlation can be quantified, for instance, by the cosine of the angle between the two vectors:
  - $\mathrm{sim}(\vec{q}, \vec{d_j}) = \cos(\vec{q}, \vec{d_j}) = \frac{\vec{q} \cdot \vec{d_j}}{|\vec{q}| \times |\vec{d_j}|} = \frac{\sum_{i=1}^{t} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^{t} w_{i,q}{}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,j}{}^2}}$

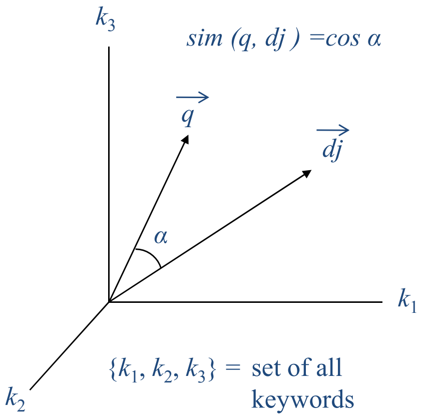


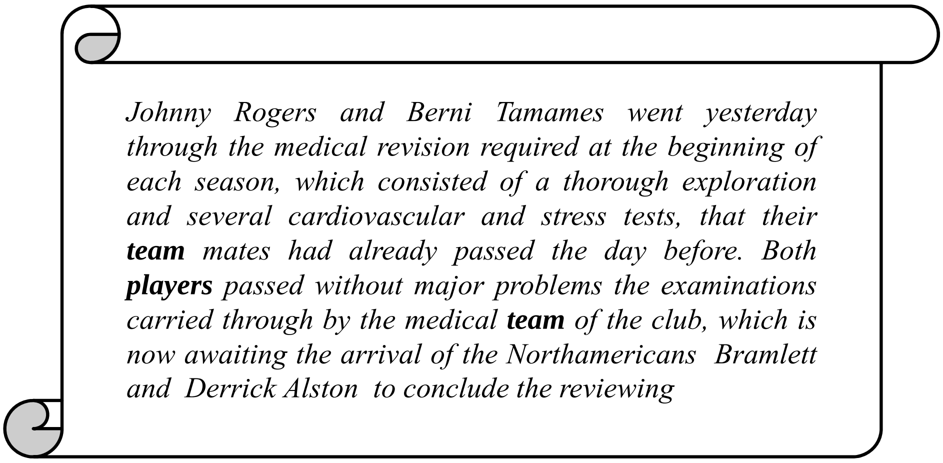
Fig 2.3 The cosine of α is adopted as $sim\ (q, d_j)$

Since $w_{i,j}$> 0 and $w_{i,q}$ > 0, sim(q, d_j) varies from 0 to 1. Thus, instead of attempting to predict whether a document is relevant or not, the VSM ranks the documents according to their degree of similarity to the query. A document might be retrieved even if it matches the query only partially.

For instance, one can establish a threshold on sim(q, d_j) and retrieve the documents with a degree of similarity above that threshold.

Example

Assume we have the query q = team, player, and the document d shown in figure 2.4, where index term weights are, let us say, w_{team}= 0.5 w_{player}= 0.8



Johnny Rogers and Berni Tamames went yesterday through the medical revision required at the beginning of each season, which consisted of a thorough exploration and several cardiovascular and stress tests, that their ***team*** *mates had already passed the day before. Both* ***players*** *passed without major problems the examinations carried through by the medical* ***team*** *of the club, which is now awaiting the arrival of the Northamericans Bramlett and Derrick Alston to conclude the reviewing*

Fig 2.4 Document represented using the vector space model

The vectors that represent the query and the document are:

$\vec{q}$ = (0, 0, …, 0, 1.0, 0, 0, …, 1.0, …, 0)

$\vec{d}$ = (0, 0, …, 0, 0.5, 0, 0, …, 0.8, …, 0)

And the similarity between them would be computed as:

$$\cos(\vec{q}, \vec{d_j}) = \frac{\vec{q} \cdot \overrightarrow{d_j}}{|\vec{q}| \times |\overrightarrow{d_j}|} = \frac{\sum_{i=1}^{t} w_{i,q} \times w_{i,\mathrm{j}}}{\sqrt{\sum_{i=1}^{t} {w_{i,q}}^2} \times \sqrt{\sum_{i=1}^{t} {w_{i,j}}^2}} = 0.97$$

The vector-space model per se does not prescribe how the values of the vector components should be computed. However, in order to effectively compute similarities and rankings, this has to be specified, which is itself a relevant issue. Term weighting is indeed a key factor in the performance of IR systems.

Extensive research and experimentation on this problem has been carried out in the past 50 years, and the proposed weighting schemes are manifold. The ultimate goal of a term weighting system is the enhancement of document retrieval effectiveness.

One of the most frequently used models for index term weighting is the **Term Frequency, Inverse Document Frequency (TF-IDF)**. This measure views the IR problem from a clustering perspective. It considers two different sets of documents: D, the complete set of information items, and R, the set of relevant information items to a query. The aim of this measure is to identify what are the features that better discriminate the elements of R from the ones outside R in D. Following this criteria, the weight of a term i in a document j w_{ij} is defined as:

$$w_{i,j} = tf_{i,j} \times idf_i = \frac{freq_{i,j}}{max_l\, freq_{l,j}} \times \log\frac{N}{n_i}$$

Where:

- N = total number of documents in the system.
- n_i = number of documents where the term t_i appears.
- $freq_{i,j}$ = frequency of the term t_i in the document d_j
- $max_l\, freq_{l,j}$ = maximum frequency of any term t_l in the document d_j

The term frequency factor, $tf_{i,j}$ aims to measure how representative is the term t_i in describing the contents of the document d_j. The inverse document frequency factor, idf_i, aims to measure how significant is the presence vs. absence of term t_i to discriminate documents from each other in the collection. The motivation behind this is that terms that appear in many documents are not useful to distinguish relevant documents from non-relevant ones.

A shortcoming of the vector space model, also present in the boolean and probabilistic models, is that index terms are assumed to be mutually independent and it is not possible to include term dependencies into the model. On the other hand, the vector space model has proved to improve retrieval performance in general respect to Boolean models. Its notion of partial matching allows retrieving documents that approximate the query, and its cosine retrieval function supports a finer order of documents based on their degree of similarity to the query.

2.3.3 Probabilistic model

The probabilistic model aims to capture the IR problem in a probabilistic framework. The fundamental idea is as follows. Given a query q and a collection of documents D, a subset R of D is assumed to exist which contains exactly the relevant documents to q (the ideal answer set). The probabilistic retrieval model then ranks documents in decreasing order of probability of belonging to this set (i.e. of being relevant to the information need), which is noted as P (R | q, d_j), where d_j is a document in D.

Following the previous notation:

- D: documents are represented as a vector of words or index terms occurring in a document. Each term in the document, that is, each pair (t_i, d_j), has a binary associated weight 1 or 0, denoting the presence or absence of the term in the document.
- Q: queries are represented by a vector of words or index terms that occur in the query. Each term in the query, that is, each pair (t_i, q) has a binary weight 1 or 0, denoting the presence or absence of the term in the query.
- F is a probabilistic model that ranks documents in order of probability of relevance to the query.
- sim measures the degree of similarity of a document d_j to a query q_i as the probability of d_j to be part of the subset R of relevant documents for q. This is measured in the probabilistic model as the odds of relevance, as given by:

$$sim(d_j, q) = \frac{P(R|d_j)}{P(\neg R|d_j)}$$

where $\neg R$ denotes the set of non relevant documents, $P(R|d_j)$ is the probability of d_j being relevant to the query q, and $P(\neg R|d_j)$ is the probability of d_j being non relevant to q.

The estimation and computation of the latter probabilities requires further elaboration, as follows (Baeza Yates & Ribeiro Neto, 1999). First, using Bayes' rule, we may write:

$$sim(d_j, q) = \frac{P(d_j|R) \times P(R)}{P(d_j|\neg R) \times P(\neg R)}$$

Assuming that P(R) and P(¬R) are the same for all documents in the collection, and considering term independence assumption, that means $P(d_j|R) = \prod_{i=1}^{t} P(t_i|R)$, we have:

$$sim(d_j, q) \sim \frac{P(d_j|R)}{P(d_j|\neg R)} \sim \frac{\prod_{i=1}^{t} P(t_i|R)}{\prod_{i=1}^{t} P(t_i|\neg R)}$$

If we consider a function g(t, d) where g(t, d) = 1 if the term t appears in the document d, and g(t, d) = 0 if the term t does not appear in the document d, the previous formula can be reformulated as:

$$sim(d_j, q) \sim \frac{(\prod_{g(t_i,d_j)=1} P(t_i|R)) \times (\prod_{g(t_i,d_j)=0} P(\neg t_i|R))}{(\prod_{g(t_i,d_j)=1} P(t_i|\neg R)) \times (\prod_{g(t_i,d_j)=0} P(\neg t_i|\neg R))}$$

The term $P(t_i|R)$ stands for the probability that the index term t_i is present in a document randomly selected from the set R. $P(\neg t_i|R)$ stands for the probability that the index term t_i is not present in a document randomly selected from the set R. The probabilities associated with the set ¬R have meanings which are analogous to the ones just described. Taking logarithms, recalling that$P(t_i|R) + P(\neg t_i|R) = 1$, and ignoring factors which are constant for all documents in the context of the same query, we can finally write:

$$sim(d_j, q) \sim \sum_{i}^{t} w_{i,q} \times w_{i,j} \times \left(log \frac{P(t_i|R)}{1 - P(t_i|R)} + log \frac{1 - P(t_i|\neg R)}{P(t_i|\neg R)} \right)$$

where $w_{i,q}$={0,1} indicates the absence/presence of the term t_i in the query q and $w_{i,j}$ ={0,1} indicates the absence/presence of the term t_i in the document d_j

Since R is unknown a priori, simplifying assumptions can be made such as:

- $P(t_i|R) = 0.5$ and constant for all index terms t_i.
- $P(t_i|\neg R) = \frac{n_i}{N}$, where n_i is the number of documents that contain t_i and N is the total number of documents.

Once an initial subset of documents V is retrieved and ranked by the probabilistic model, the probabilities can be refined to:

- $P(t_i|R) = \frac{|V_i|}{|V|}$, where V_i is the set of retrieved documents containing t_i.
- $P(t_i|\neg R) = \frac{n_i - |V_i|}{N - |V|}$, by considering that the non-retrieved documents are not relevant.

Following this process recursively we get:

- $P(t_i|R) = \frac{|V_i|+\frac{n_i}{N}}{|V|+1}$
- $P(t_i|\neg R) = \frac{n_i - |V_i|+\frac{n_i}{N}}{N-|V|+1}$

Example

Documents	Set of index terms						
	Cold	**Day**	**Eat**	**Hot**	**Meal**	**Pizza**	**drink**
d1	1	0	1	0	0	0	0
d2	0	1	1	1	0	0	0
d3	0	0	1	0	1	1	1
d4	0	1	0	0	1	0	0

Table 2.1 Probabilistic model data example[4]

If we have $V = \{d_1, d_2\}$ and we want to compute the relevance of d_1

$$P(\text{Cold}|R) = \frac{|V_i|+\frac{n_i}{N}}{|V|+1} = \frac{1+\frac{1}{4}}{2+1} = 0.42 \quad P(Cold|\neg R) = \frac{n_i - |V_i|+\frac{n_i}{N}}{N-|V|+1} = \frac{1-1+\frac{1}{4}}{4-2+1} = 0.08$$

$$P(\text{Eat}|R) = \frac{|V_i|+\frac{n_i}{N}}{|V|+1} = \frac{2+\frac{3}{4}}{2+1} = 0.92 \quad P(Eat|\neg R) = \frac{n_i - |V_i|+\frac{n_i}{N}}{N-|V|+1} = \frac{3-2+\frac{3}{4}}{4-2+1} = 0.58$$

$$\text{sim}(d_{1,}\ q) \sim \log\left(\frac{0.42}{0.58}\right) + \log\left(\frac{0.92}{0.08}\right) + \log\left(\frac{0.92}{0.08}\right) + \log\left(\frac{0.42}{0.58}\right) = 1.84$$

Shortcomings of the probabilistic models include: (1) the need to guess the initial separation of documents into relevant and non-relevant sets; (2) the fact that the classic model does not take into account the frequency of index terms inside documents (i.e., all weights are binary).

Despite these shortcomings, variations of the probabilistic model have lead to the development of one of the most successful ranking models, BM25 (Robertson & Sparck Jones, 1976) (Sparck Jones, Walker, & Robertson, 2000). The first system to incorporate this function was the Okapi information retrieval system, implemented at London's City University in the 1980s and 1990s. This ranking methodology takes into account the present/absence of relevant information and incorporates a document-specific component, which measures term frequencies and documents lengths.

2.3.4 Additional models

Over the years, alternative modeling paradigms have been proposed. Among those models we can highlight: the fuzzy and the extended Boolean models, the generalized vector model, the neural network models, etc. An introduction to these models can be found in (Baeza Yates & Ribeiro Neto, 1999). More recently, so-called Language Models have become popular and widely studied in the IR

[4] A value of 1.0 indicates that the index term is present in the document

research field, because of their good performance and the fact that they unify term weighting and result ranking in a single model with probabilistic foundation (Ponte & Croft, 1998).

2.4 Information Retrieval evaluation

With the continued information explosion, including the emergence of the internet and digital library initiatives, IR performance has become increasingly critical. In the current commercial competition designers, developers, vendors and sales representatives of new information products need to carefully study whether and how do their products offer competitive advantages. There are broadly three types of evaluation of information retrieval systems (Baeza Yates & Ribeiro Neto, 1999): The first one is functional evaluation, in which the specified system functionalities are tested one by one. The second one is the performance evaluation. The most common measures of system performance are time and space (the shorter the response time, the smaller the space used, the better the system is considered to be). The third one is the *retrieval performance evaluation*. This evaluation assesses how well the IR system satisfies the information need of its users. There are two broad classes of retrieval performance evaluation: a) user-based retrieval performance evaluation and b) *system-based retrieval performance evaluation*. The first one measures the user′s satisfaction with the system, while the second one focuses on how well the system can rank documents. User-based evaluation is in principle, much more informative and useful but is extremely expensive and difficult. On the other hand, system-based retrieval performance evaluation is, by design, an abstraction of the retrieval process that allows experiments to control some of the variables that affect retrieval performance thus increasing the power of comparative experiments. They are much less expensive than user-based evaluations while providing more diagnostic information regarding system behavior.

System-based retrieval performance evaluation is based on the Cranfield evaluation paradigm (Cleverdon, 1967) (Cleverdon, 1991). In this paradigm, researchers perform experiments on test collections to compare the relative effectiveness of different retrieval approaches using several evaluation measures. The test reference collection generally consists of a collection of documents, a set of sample queries, and a set of relevant documents (judgments), manually identified for each query. Given a retrieval strategy S, for each query the evaluation measure quantifies the similarity between the set of documents retrieved by S and the set of known relevant documents. This provides an estimation of the goodness of the retrieval strategy.

The following sections show an overview of the most common evaluation metrics and tests collections used for *system-based retrieval performance evaluation*.

2.4.1 Evaluation metrics

An important amount of metrics have been developed to evaluate retrieval models, and this is actually an active and interesting area of research. However, none of those metrics is completely satisfactory, because retrieval performance evaluation measures are user-dependent and multidimensional, while the result of these measures is a single value. The evaluation metrics are performed by using a document collection, a set of topics describing a user’s information needs and a set of relevance judgments, indicating, for each topic, which documents are (manually annotated) relevant for each topic. This judgment is usually binary (relevant or not relevant) and generally incomplete, as not every document is classified into the relevant and non-relevant classes.

Two basic and probably the most common retrieval performance evaluation metrics are *Precision* and *Recall*. Consider an example query q and its set of relevant documents R. Let A be the set of documents returned for q by a given retrieval strategy under evaluation, and let Ra be the documents in the intersection of R and A, i.e. the relevant documents in the answer set. Recall and precision are defined as follows:

- **Recall** – is the fraction of the relevant documents which has been retrieved (|Ra|/|R|).
- **Precision** – is the fraction of the retrieved documents which are relevant (|Ra|/|A|).

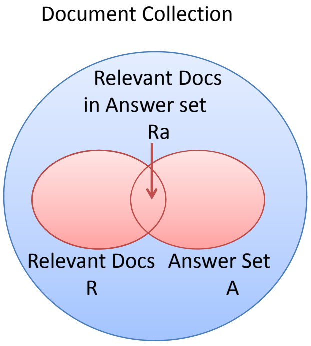


Fig 2.5 Precision and recall for a given query

The values of recall and precision are between 0 and 1. The higher the recall value, the better the retrieval performance. Similarly, the higher the precision value, the better the retrieval performance. Besides the global precision value for a whole result list output by a system, it is common to measure precision at specific positions of the ranking, which is commonly denoted by **precision@n**, n being the rank position.

Note that precision and recall are set-based measures. They evaluate the quality of an unordered set of retrieved documents. To evaluate ranked lists, recall-precision curves are used. For those cases, **Precision at 11 standard recall levels** is measured. Each recall-precision point is computed by calculating the precision at the specified recall cutoff value. For the rest of recall values, the precision is interpolated as:

$$P(r_j) = \max r_j \leq r \leq r_{j+1} P(r)$$

Example

Assume the set of relevant documents to a query q is Rq = {d3, d5, d9, d25, d39, d44, d56, d71, d89, d123}, and the ranking by a specific retrieval system is:

1 d123 *
2 d84
3 d56 *
4 d6
5 d8
6 d9 *
7 d511
8 d129
9 d187
10 d25 *
11 d38
12 d48
13 d250
14 d113
15 d3 *

where (*) indicates the relevant documents to the query q.

Since there are 10 relevant documents, and the first document d123 is considered relevant, we have a 100% of precision at 10% of recall. The 20% of recall is obtained with the third document, d56. Therefore we get abound 66% of precision at 20% of recall. In this example the precision at levels of recall higher than 50% drops to 0 because not all relevant documents have been retrieved.

Recall	Precision
0.00	1
0.10	1
0.20	0.66
0.30	0.5
0.40	0.4
0.50	0.33
0.60	0
0.70	0
0.80	0
0.90	0
1.00	0

Table 2.2 Precision at 11 standard recall levels over all relevant documents

Based on this recall-precision curve, we can see that: the recall and precision values for an ideal search would both be 1. However, in practice, there is always a trade-off between recall and precision. For example, as recall approaches 1, precision tends to drop to 0, which means the search returns most relevant docs but also includes lots of non relevant ones; when precision tends to 1, recall value will approximate to 0, which means the search returns relevant documents but misses many useful documents too.

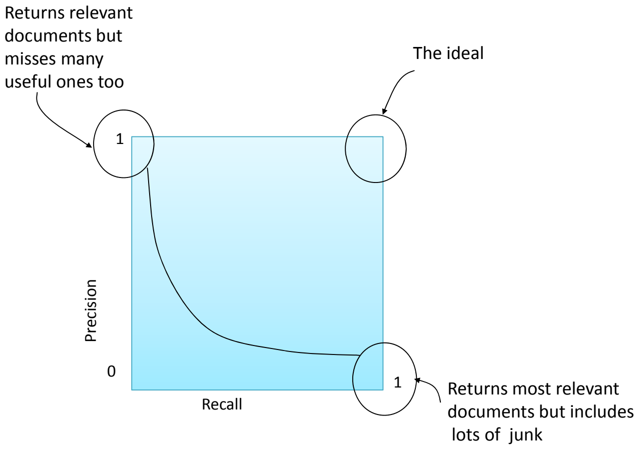

Returns relevant documents but misses many useful ones too
The ideal
1
Precision
0
Recall
1
Returns most relevant documents but includes lots of junk

Fig 2.6 Trade-off between recall and precision

As a global estimate of performance across multiple recall levels, it is standard to use **Average Precision (AP)**. This measure is defined as the arithmetic mean of the precision at all the positions in the ranking where a relevant document occurs. To get an average precision of 1.0, a retrieval system must retrieve all relevant documents (i.e., recall = 1.0) and rank them all in the topmost positions, without mix of irrelevant documents (i.e. precision = 1.0 at all positions down to the last relevant document). This measure can be averaged across a set of queries, in which defines the **Mean Average Precision (MAP)**.

Another overall performance measure is **R-precision**. It computes the precision when |R| documents have been retrieved, R being the set of all relevant documents for the query. In the previous example the value of R-Precision is 0.4 because the number of relevant documents for the query is 10 and there are 4 relevant documents among the first 10 documents in the ranking. The R-precision measure is a useful parameter for observing the behavior of an algorithm for each individual query in an experiment.

Even though recall and precision are the most popular measures to evaluate the retrieval performance; they suffer from a number of drawbacks that limit their usefulness, or make them difficult to apply in certain cases:

- The proper estimation of maximum recall for a query requires detailed knowledge of all the documents in the collection.
- The use of recall and precision individually does not provide enough information, and a combination of both measures is generally required to perform the evaluation.
- The use of recall and precision evaluation measures is insufficient for interactive systems where the user specifies his information need through a series of interactive steps with the system.
- The use of recall and precision might be inadequate for systems where the final ranking of documents requires a weak ordering and not a linear ordering.

Because of this set of drawbacks, alternative evaluation measures have been proposed. Some of them such as the **Harmonic Mean** (Shaw, Burgin, & Howell, 1997) or the **E-Measure** (Rijsbergen, 1979) propose combinations of recall and precision. While the Harmonic Mean reflects a compromise between recall and precision, the E-Measure allows the user to specify whether he is more interested in recall or in precision. Other measures such as **bpref** (Buckley & Voorhees, 2004) are devised for situations where there is not detailed knowledge of all the documents in the collections and the relevance judgments are known to be far from complete.

There is also a set of user-oriented evaluation measures. These measures are based on the assumption that the set of relevant documents for a query is not the same for each user. For instance, the **coverage ratio** (Korfhage, 1997) which measures the fraction of relevant documents retrieved that is known to the user and, **the novelty ratio** (Korfhage, 1997) which measures the fraction of relevant documents retrieved that is unknown to the user.

The list of performance metrics described here is not exhaustive, as this is in fact, as pointed out earlier in this section, an active open field of research. Other popular metrics not covered in this section include for instance: expected search length, satisfaction, frustration, etc, which are described here (Korfhage, 1997).

2.4.2 Reference collections

System-based retrieval performance evaluation uses test collections as a mechanism for comparing system performance. A test collection consists of three distinct components:

- A representative set of documents.
- A representative set of topics or queries.
- A set of relevant judgments (or lists of relevant documents for each topic)

System-based retrieval performance evaluation was initially based on the Cranfield experiments (Cleverdon, 1967) (Cleverdon, 1991). In these experiments three assumptions were made:

- Relevance can be approximated by topical similarity. This assumption has several implications: a) all relevant documents are equally desirable, b) the relevance of one document is independent of the relevance of any other document, and c) the user information need is static.
- A single set of judgements for a topic is representative of the user population.
- The list of relevant documents for each topic is complete (all relevant documents are known).

In general, these assumptions do not strictly hold in practice. Relevance is inherently subjective and judgments are known to vary between individuals, context and time. The most widely reported retrieval effectiveness measures are based on binary relevance judgments, but assessors generally report greater confidence in their judgments when they can express the relevance degree of an item on a multi-valued scale. On the other hand, early attempts at building IR test collections exhaustively judged the relevance of every document to every topic. However, for large collections and large numbers of topics (needed to achieve stable measures), providing complete relevant judgements is not feasible. A widely used alternative is *pooled assessment*, in which top-ranked documents from many systems are judged, and unjudged documents are treated as if they were not relevant.

Because the assumptions upon which the Cranfield paradigm is based are not strictly true, the evaluation of retrieval systems is considered a noisy process (Voorhees E. , 2001). The primary consequence of this noise is the fact that evaluation scores computed for a test collection are valid only in comparison to scores computed for other runs using the exact same collection. A second consequence of this noise is that there is an (unknown) amount of error when comparing two systems on the same collection, This error can be reduced by repeating the whole experiment (different sets of topics /judgements) multiple times.

Even though the Cranfield paradigm is considered a noisy methodology, it is still the most popular, and standardized way of evaluating IR systems. The following sections describe some of the most popular Cranfield-based test collections:

- The Cranfield test collection.
- The CACM collection.
- The TREC collection

2.4.2.1 The Cranfield and CACM collections

In 1967, the Cranfield studies (Cleverdon, 1967) emphasized the importance of creating reference test collections and using these for comparative evaluation. The Cranfield collection, created in 1960s, contains approximately 1400 documents and 225 queries. It was built with the aim of testing hypothesis about the manual indexing of documents. The queries are not "natural" or random user requests but were specifically constructed considering the documents in the collection so that there were a significant number of relevant documents in each request.

The CACM (Communications of the Association of Computing Machinery) collection was created in 1983. It was built to investigate the interaction between textual and bibliographic data. It has 64 independent requests gathered from students and an independent test collection of 3204 articles, but nearly 50% of the articles are just a title.

Even though these collections might support somewhat challenging tests when they were used for the first time, they are relatively small and therefore they misrepresent several important issues of large bibliographical environments: performance in large full-text search, abilities to operate in real-world conditions, etc. Another important limitation of these collections is that they were built to support specific experimental purposes and therefore, its reuse is sometimes complicated.

2.4.2.2 The TREC collection

The TIPSTER/TREC collection is usually considered to be the reference test collection in IR nowadays. It is the result of a project launched in 1991 by the National Institute of Standards and Technology (NIST). Its purpose was to support research within the information retrieval community by providing the necessary infrastructure for large-scale evaluation of text retrieval methodologies. This was driven as a series of annual workshops focused on a list of different IR research areas, or *tracks.*

The TIPSTER test design was based on traditional information testing models, involving a test collection of documents, user requests (called topics) and relevant judgments.

- *The document collection*: A very large collection was needed to test the ability of the algorithms to handle huge numbers of full-text documents. The documents needed to cover many different subject areas in order to test the domain independence of the algorithms. Additionally, the collection needed to cover the different types of documents (varied length, different writing style, different level of editing, different vocabulary, etc.). As a final requirement the documents needed to cover different years to show the effects of document dates. The TREC collection has been growing steadily over the years. At TREC 3 the collection size was 2 GB while at TREC 6 it has grown up to 5.8 GB. The documents come from sources such as: Wall Street Journal, Associate Press (news wire), Computer Selects (articles), Federal Register, US DOE publications (abstracts), San Jose Mercury News, US Patents, Financial times, Congressional Record and LA times. Documents from all subcollections are tagged with SGML to allow easy parsing.
- *The Information Requests (topics)*: The topics (requests) were created to be quite specific, but included both broad and narrow searching needs. The task of converting the information request (topic) into a system query must be done by the system itself and it is considered to be

an integral part of the evaluation procedure. The number of topics prepared for the first six TREC conferences grew up to 350.

- *The relevance assessments (judgements):* At TREC conferences, the set of relevant documents for each example information request (topic) is obtained from a pool of possible relevant documents. This pool is created by taking the top K documents (usually K=100) in the rankings generated by the various participating retrieval systems. The documents in the pool are then showed to human assessors who ultimately decided on the relevance of each document.
- *The Tasks:* The first eight TREC cycles were centred on two main tasks: a) the ad hoc task and b) the routing task. In the first one it is assumed that new requests are asked over a fixed set of data. This is represented by new topics for known documents. In the second one it is assumed that the same requests are always being followed but new data is searched. This is represented by using known topics and known relevant documents for those topics, but new data for testing. Starting at the TREC 4 conference, new secondary tasks, besides the ad hoc and routing tasks, were introduced as new research needs were identified. The current complete list for 2008 and previous years includes[5]:
    - Blog Track: to explore information seeking behavior in the blogosphere.
    - Enterprise Track: to study search over the data of an organization to complete some task.
    - Genomics Track: to study the retrieval of genomic data, not just gene sequences but also supporting documentation such as research papers, lab reports, etc.
    - Legal Track: to develop search technology that meets the needs of lawyers to engage in effective discovery in digital document collections.
    - Million Query Track: to test the hypothesis that a test collection built from many very incompletely judged topics is a better tool than one built using traditional TREC collection pooling. New for 2007.
    - Relevance Feedback Track: to explore the effects of different factors on the success of relevance feedback
    - Question Answering Track: to achieve more information retrieval than just document retrieval by answering factoid, list and definition-style questions.
    - Spam Track: to provide a standard evaluation of current and proposed spam filtering approaches.
    - Cross-Language Track: to investigate the ability of retrieval systems to find documents topically regardless of source language.
    - Filtering Track: to binarily decide retrieval of new incoming documents given a stable information need.

[55] http://trec.nist.gov/

- HARD Track: to achieve High Accuracy Retrieval from Documents by leveraging additional information about the searcher and/or the search context.
- Interactive Track: to study user interaction with text retrieval systems.
- Novelty Track: to investigate systems' abilities to locate new (i.e., non-redundant) information.
- Robust Retrieval Track: to focus on individual topic effectiveness.
- Terabyte Track: to investigate whether/how the IR community can scale traditional IR test-collection-based evaluation to significantly large collections.
- Video Track: to research in automatic segmentation, indexing, and content-based retrieval of digital video.
- Web Track: to search on a document set that is a snapshot of the World Wide Web.

The Web track evaluation benchmark, and more specifically the TREC WT10g document collection, is adapted to perform the evaluation of the semantic retrieval system implemented in this work. Given the magnitude and importance of this document set, we dedicate a separate section next to briefly describe this large-scale Web-based collection.

2.4.2.3 The TREC WT10g collection

The purpose of the Web Track was to build a test collection that mimics the retrieval environment of the WWW as closely as possible. In the initial years of the Web track, the TREC VLC, VLC2 and WT2G Web datasets were collected. Experimental work in TREC 7 and TREC 8 editions demonstrated that these datasets were not appropriate to simulate the salient properties of real Web search, and therefore to properly evaluate the systems. Too few relevance judgments were available for VLC collections to support the pooling assumption that unjudged documents can safely be considered irrelevant. While WT2g addressed this limitation, it was very small and contained very few inter-server links. As a result for TREC 9 and TREC 2001 editions, efforts were concentrated on the engineering of a new corpus, to be known as WT10g collection. This corpus was constructed to support the following characteristics:

- Model real Web search by means of: a) a sufficiently large and representative document set, b) a large set of representative Web queries and c) a corresponding set of “sufficiently complete” relevant judgments.
- Enable meaningful evaluation of hyperlinked-based retrieval methods.
- Support experimentation with server selection and result merging algorithms for distributed IR.
- Be neither too large nor too “messy” to discourage its use. Large scale use of the corpus is necessary for the success of the pooling method in building up reusable relevance judgements.

(Bailey, Craswell, & Hawking, 2003) describes the construction of this collection known as WT10g and used in the TREC 9 and TREC 2001 Web tracks. The collection is about 10GB in size, and contains 1.69 million Web pages. Their goal was to create a testbed for realistic and reproducible experiments on Web documents with traditional, distributed and hyperlink-based retrieval algorithms. The construction began with VLC2, a 100GB subset of a 1997 crawl by the Internet Archive. From documents were selected by a process designed to maximize inter-server connectivity, retain as many pages as possible from each represented server, incorporate documents which are likely to be relevant to a wide variety of queries, and exhibit a realistic distribution of server sizes. This process is described in detail in (Bailey, Craswell, & Hawking, 2003). The properties of the resulting collection were measured according to the mean in- and out-links per server, the fraction of connected servers in the collection, and server "relevance", measured using a large query set. Under the TREC experience with this collection it was observed that, even though the WT10G collection does contain exploitable link evidence, it comprises less than 0.1% of the pages reportedly indexed by major search engines.

The standard procedure for topic creation was also tweaked to create the topics for the Web track. Participants in the Web track were concerned that the queries that users type in current Web search engines are quite different from standard TREC topic statements. However, if participants were given only the literal queries submitted to a Web search engine, they would not know the criteria by which documents should be judged. As a compromise, standard TREC topic statements were retrofitted around Web queries. TREC 2001 topics were obtained from MSN search logs. Each assessor selected a query and developed a description and a narrative for that query. The assessors were instructed that the original query might be ambiguous ("cats"), and they were to develop a description and a narrative that were consistent with any interpretation of the original query (e.g., "Where is the musical of Cats playing?"). While the description and narrative fields use correct American English, the title field may contain spelling errors (for TREC 9 topics) and punctuation and grammar mistakes (for TREC 9 and TREC 2001 topics).

```
<top>
    <num> Number: 501
    <title>  deduction and induction in English?
    <desc> Description:
    What is the difference between deduction and induction in the process of reasoning?
    <narr> Narrative:
    A relevant document will contrast inductive and deductive reasoning.
    A document that discusses only one or the other is not relevant.
</top>
```

Fig 2.7 Example of a TREC 2001 topic

2.5 Summary

In this chapter we have given a briefly overview of IR technologies, theories and systems. The architecture of a general IR system was introduced, with special emphasis on the different steps involved in the IR task: indexing, query processing, searching and ranking.

Focusing on the problem of ranking, classic IR models have been introduced: a) *the Boolean model*, where documents are represented as sets of words or phrases; b) *the VSM*, where documents and queries are represented as vectors, matrices or tuples and c) *the Probabilistic model*, where the process of document retrieval is treated as a probabilistic inference, generally based on theorems like the Bayes' theorem. We pay special attention to the **VSM**, which is the one that has been adapted for the semantic retrieval model proposed in this thesis, as will be described in Chapter 5.

Another key issue we have focused on here is the evaluation of systems and models. This chapter reviews some of the most relevant IR evaluation measures, procedures and collections. We have explained metrics such as **recall** and **precision**. We have highlighted the **Cranfield methodology** (Cleverdon C. , 1967) (Cleverdon, 1991), as the most widely applied evaluation approach. It uses test collections as a mechanism to compare the performance of the different search systems. A test collection consists of three distinct components: a) a set of documents, b) a representative set of topics or queries and, c) a set of relevant judgments (or lists of relevant documents for each topic). We have briefly described popular collections such as Cranfield, CACM, and TREC collection. Special attention was paid to the **TREC WT10g collection**, which is the one used in this thesis to conduct the large-scale experiments of the proposed semantic retrieval model, as will be shown in detail in section 6.3.

Chapter 3

Semantic-based knowledge technologies

In this chapter, we survey the development towards semantic-intensive knowledge technologies in the last decade with the aim to automate tasks using software that substitutes human knowledge (section 3.1). We introduce and revise the different problems of semantic-based knowledge representation (section 3.2), acquisition (section 3.3), annotation (section 3.4) and evaluation (section 3.5), stressing the ontologies as the cornerstone of semantic technologies.

3.1 Motivation

Barely a decade after its conception, the World Wide Web (WWW), has become a commodity we use on a daily basis, comparable to other very important media such as the radio, TV or the telephone, but surpassing these in many aspects. The Web today is an extremely versatile and economic medium to perform tasks such as communication, trade and business, leisure and entertainment, access to information and services, culture dissemination, etc. In parallel with the spectacular growth of the Web, its technologies have experienced an extraordinarily fast evolution. Since the first basic technologies: HTML[6] and HTTP[7], up to our days, with the emergence of technologies such as CGI[8], Java[9], JavaScript[10], ASP[11], JSP[12], PHP[13], Flash[14], j2ee[15], XML[16], to name some of the best known, allowing a better and more powerful Web. These changes affect and are influenced by the transformation of the WWW. The generation of dynamic pages, the link to databases, the increased interac-

[6] http://www.w3.org/MarkUp/
[7] http://www.w3.org/Protocosl/
[8] http://hoohoo.ncsa.uiuc.edu/cgi/
[9] http://java.sun.com/
[10] http://www.mozilla.org/js/
[11] http://www.asp.net/
[12] http://java.sun.com/products/jsp/
[13] http://php.apache.org/
[14] http://www.macromedia.com/
[15] http://java.sun.com/j2ee/
[16] http://www.w3.org/XML/

tivity with the user, the design of the Web as a universal platform for the deployment of applications, etc., are some of the most prominent trends in recent years.

The Web is growing so fast that it is impossible to give a precise and absolute measure of its current dimensions. The latest studies estimate its size would comprise in the range of 20 billion documents according to the Yahoo statistics extracted in 2005[17]. Today almost everything is represented in one way or another on the Web, and with the help of a good search engine we can find information about virtually anything we may need. The Web can be said, to have in many ways become, a universal encyclopedia of human knowledge. Furthermore, the Web allows us to take care of the widest variety of daily life activities and needs with unprecedented efficiency, economy, and comfort: we can buy all kinds of products and services, manage a bank account, find a restaurant, read newspapers, locate a person, access maps, etc.

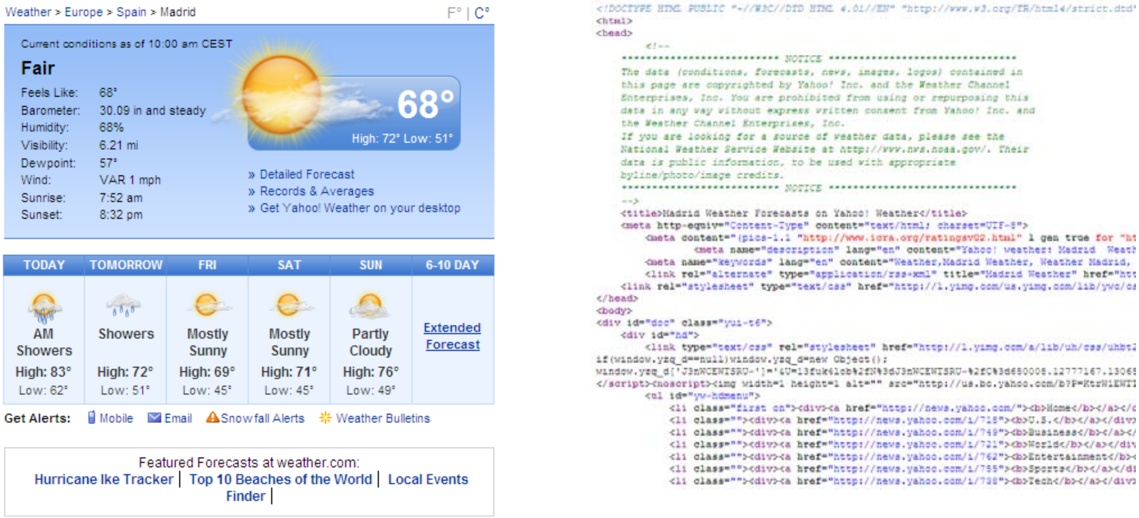
The enormous size that the Web has reached is one of the keys to its success, but it also makes some tasks very time-consuming and tedious for users (e.g., find the optimal planning, including transport, accommodation, etc, among all possible options, to travel under certain conditions). On the other hand, the development of programs that take care of these tasks on our behalf often involves considerable complexity, as it is very difficult to reproduce (and maintain) in a machine, a person's ability to understand the processes and information as they are currently supported and encoded.

The effectiveness of current search engines has also its limits. For example, if we ask questions such as "*list of companies that trade on Nasdaq*"[18] we get a list of articles about the American Stock Exchange Nasdaq, about its history, and about the incidents of some companies related to Nasdaq. However, among the first top 10 results, we do not get any page containing the Nasdaq index of companies.

All these examples illustrate a common limitation. The contents and services of the Web use formats (such as HTML) that can be understood by humans, but not by machines. Fig 3.1 represents this situation with a simplified version of a meteorological Web Page. While the presentation of the data in the browser is easily interpretable by humans, it is nearly impossible for a computer to understand the current temperature, the forecast for the following days, and other semantics of the document. This is due to the fact that semantics and style format tags are interspersed.

[17] http://www.ysearchblog.com/archives/000172.html

[18] These results have been obtained at the time of writing with the Google search engine (www.google.com). The fast evolution of this search engine may entail different results for the same queries in the next future.



Weather > Europe > Spain > Madrid F° | C°

Current conditions as of 10:00 am CEST

Fair

Feels Like:	68°
Barometer:	30.09 in and steady
Humidity:	68%
Visibility:	6.21 mi
Dewpoint:	57°
Wind:	VAR 1 mph
Sunrise:	7:52 am
Sunset:	8:32 pm

68°
High: 72° Low: 51°

» Detailed Forecast
» Records & Averages
» Get Yahoo! Weather on your desktop

TODAY	TOMORROW	FRI	SAT	SUN	6-10 DAY
AM Showers	Showers	Mostly Sunny	Mostly Sunny	Partly Cloudy	Extended Forecast
High: 83°	High: 72°	High: 69°	High: 71°	High: 76°	
Low: 62°	Low: 51°	Low: 45°	Low: 45°	Low: 49°	

Get Alerts: Mobile Email Snowfall Alerts Weather Bulletins

Featured Forecasts at weather.com:
Hurricane Ike Tracker | Top 10 Beaches of the World | Local Events Finder |

The Web seen by a human

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd'
<html>
<head>
        <!--
    ************************* NOTICE ******************************
    The data (conditions, forecasts, news, images, logos) contained in
    this page are copyrighted by Yahoo! Inc. and the Weather Channel
    Enterprises, Inc. You are prohibited from using or repurposing this
    data in any way without express written consent from Yahoo! Inc. and
    the Weather Channel Enterprises, Inc.
    If you are looking for a source of weather data, please see the
    National Weather Service Website at http://www.nws.noaa.gov/. Their
    data is public information, to be used with appropriate
    byline/photo/image credits.
    ************************* NOTICE ******************************
    -->
    <title>Madrid Weather Forecasts on Yahoo! Weather</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta content="(pics-1.1 "http://www.icra.org/ratingsv02.html" l gen true for "ht
                <meta name="description" lang="en" content="Yahoo! weather: Madrid  Weath
        <meta name="keywords" lang="en" content="Weather,Madrid Weather, Weather Madrid,
        <link rel="alternate" type="application/rss+xml" title="Madrid Weather" href="htt
    <link rel="stylesheet" type="text/css" href="http://l.yimg.com/us.yimg.com/lib/ywc/cs
</head>
<body>
<div id="doc" class="yui-t6">
    <div id="hd">
        <link type="text/css" rel="stylesheet" href="http://l.yimg.com/a/lib/uh/css/uhbt2
if(window.yzq_d==null)window.yzq_d=new Object();
window.yzq_d['J3nWCEWISRU-']='&U=13fuk6lcb%2fN%3dJ3nWCEWISRU-%2fC%3d650006.12777167.13065
</script><noscript><img width=1 height=1 alt="" src="http://us.bc.yahoo.com/b?P=KtrW1EWTI
        <ul id="yw-hdmenu">
            <li class="first on"><div><a href="http://news.yahoo.com/"><b>Home</b></a></d
            <li class=""><div><a href="http://news.yahoo.com/i/716"><b>U.S.</b></a></div>
            <li class=""><div><a href="http://news.yahoo.com/i/749"><b>Business</b></a></
            <li class=""><div><a href="http://news.yahoo.com/i/721"><b>World</b></a></div
            <li class=""><div><a href="http://news.yahoo.com/i/762"><b>Entertainment</b><
            <li class=""><div><a href="http://news.yahoo.com/i/755"><b>Sports</b></a></di
            <li class=""><div><a href="http://news.yahoo.com/i/738"><b>Tech</b></a></div>
```

The Web seen by a machine

Fig 3.1 Differences between humans and machines view of the same Web page

In these conditions it is very difficult to automate tasks by software acting as human knowledge-base surrogate. A program can point the user to a specific Web site. It may build, transport, process and provide information, but it does not know what this information means, and therefore, its ability to perform autonomous actions is very limited.

Semantic technologies, aim to overcome this limitation by **introducing explicit descriptions**, as well as the internal structure and the overall structure of content and services. Fig 3.2 illustrates this proposal. The first image is an example of the current Web content representation formed by nodes of a single type (HTML pages), and edges (hyperlinks) equally undifferentiated. Hence, for example, there is no distinction between a personal Web page of a software developer and the Web site of a software application. The second image is an example of the Web content representation using semantic technologies. Every node (resource) is assigned a specific type/class/category (person, software, document, place, etc.), and the edges represent relations explicitly differentiated (software – document, software – software, document – creator, etc.)

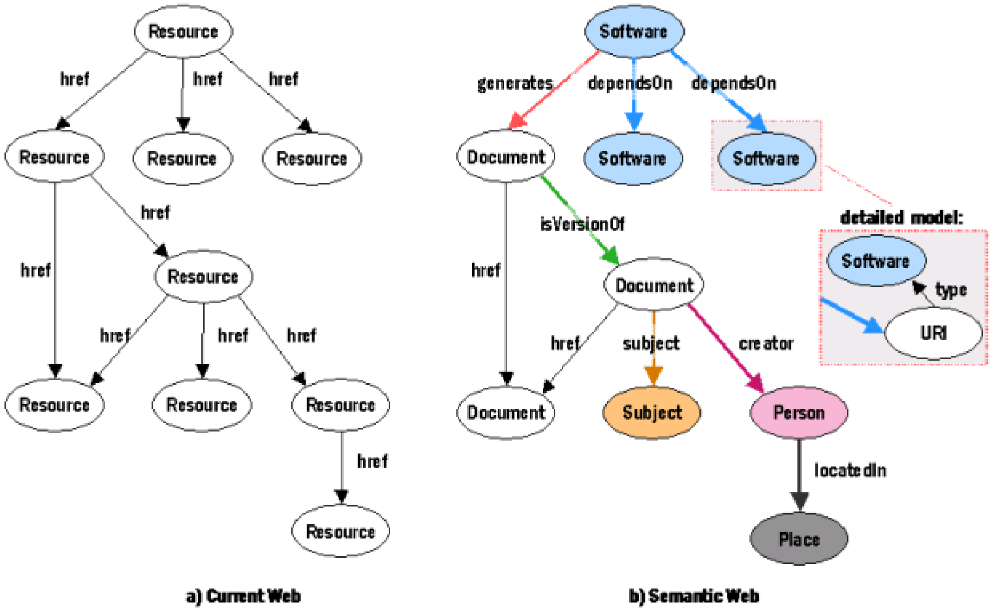


Fig 3.2 Current Web content structure vs. Semantic Web content structure[19]

To introduce explicit descriptions of contents and services semantic technologies use different knowledge representations. Among the most popular ones we may highlight the concept of "ontology" from the Artificial Intelligence (AI) field. Briefly explained, an ontology is a hierarchy of concepts with attributes and relations that defines an agreed terminology to describe semantic networks of interrelated information units. An ontology provides a vocabulary of classes and properties to describe a domain, emphasizing the sharing of knowledge and the consensus about its representation. For instance, an ontology about *Computer applications* could include classes such as *Software*, *Document*, *Person*, and properties (relations) like Person *creator* of a document, software *depends on* software, or software *generates* document.

The goal is then to describe services and contents by a network of nodes typified and interconnected through classes and properties defined in shared ontologies. Thus, for example, once an ontology about computer applications had been created, a virtual company could organize its contents defining instances of applications, developers, documents, etc. A software agent browsing a network like that might recognize the different information units, obtain specific data or reason about complex relations. At that point, we could distinguish between a software x that *is called* by a software y, and a software x that *makes a call* to a software y.

The Web not only provides access to contents, but also offers interaction and services (buying a movie, booking a flight, making a bank transfer, etc.). These services are an important research line for semantic technologies, which propose the description of functionalities and procedures to represent Web services: their inputs and outputs, the constraints to satisfy for their execution, the effects that they produce, or the steps to follow when dealing with complex services. These machine-

[19] Extracted from http://www.w3.org/2001/12/semweb-fin/current-vs-sw.png

processable descriptions would allow the automation, discovering, composition, and execution of services, as well as the communications among them.

Semantic technologies were built to contribute to the realization of the so-called Semantic Web (SW) vision. The SW emerged at the end of the 90 (Deshpande & Karypis, 2004) and was promoted by the Web inventor and president of the W3C consortium. The SW proposes to introduce machine-processable semantic representations into the Web, allowing programs to read, understand and use data over the WWW and accomplish useful goals for users. The SW aims to develop a more cohesive Web, where it is easier to find, share and integrate information and services. Even though at the time of writing, the full vision of the SW has yet to be fully accomplished (Cardoso, 2007), semantic technologies constitute a clear step forward. Semantic technologies provide an abstraction layer above current IT technologies that enables bridging and interconnection of data, content, and processes. They can be thought of as a new level of depth that provides far more intelligent, capable, relevant, and responsive interaction than with information technologies alone.

The application of semantic technologies entails some difficulties and problems such as: the **semantic knowledge representation**, which refers to how to describe and represent semantic information in the best way to be understood for applications, the **semantic knowledge acquisition**, that refers to how to obtain the semantics used to describe contents and services, the **semantic knowledge annotation** that refers to how the meanings and information conveyed by contents and services can be formally described, albeit to an incomplete extent, with metadata (semantic knowledge), and the **semantic knowledge evaluation** which refers to how to determine the quality of the semantic knowledge.

3.2 Semantic knowledge representation

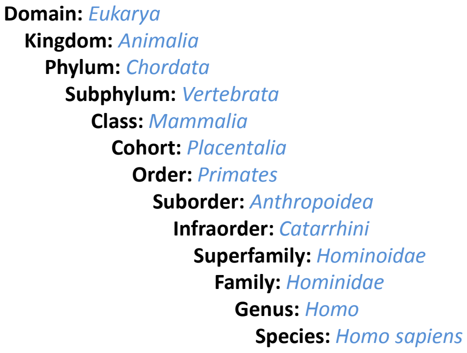
Semantic knowledge representation is the study of how knowledge about the world can be represented. It is commonly used to refer to representations intended for processing by computers, and in particular, for representations consisting of explicit objects (e.g., the class of all elephants, or Clyde a certain individual), and of assertions or claims about them (e.g., Clyde is an elephant, or all elephants are grey). Representing knowledge in such explicit form enables computers to draw conclusions from knowledge already stored (e.g., Clyde is grey).

Important questions in semantic knowledge representation include the tradeoffs between representational adequacy, fidelity, acquisition cost and computational cost. Considering these tradeoffs four different semantic knowledge representations can be identified in the literature, from the less semantically representative, the bag of words, to the most complete in terms of semantic knowledge representation, the ontology.

- **Bags of words**: uncategorized terms.
- **Taxonomies:** categories + hierarchical relations.
- **Thesauri:** categories + fixed hierarchical and associative relations.
- **Ontologies:** classes + instances + arbitrary semantic relations + rules.

The **bag-of-words** model is a simplifying assumption where knowledge is represented as an unordered collection of words, commonly known as tags, disregarding grammar and even word order if exist. This knowledge representation has been commonly used in recent years by the Web 2.0 community to categorize content such as Web pages (delicious[20]), photographs (flirk[21]), etc., through collaborative efforts from the online community.

Following the definition of Daconta (Daconta, Obrst, & Smith, The Semantic Web: A guide to the future of XML, Web Services and Knowledge Management, 2003) **Taxonomy** is defined as: "The classification of information entities in the form of a hierarchy, according to the presumed relationships of the real-world entities that they represent". A taxonomy classifies terms hierarchically, using the father-son, is-a, or type-of relationship. Indeed, taxonomies allow only the father-son relationship, dismissing other types of relations such as: part-of, cause-effect, association and location. Furthermore, taxonomies do not permit defining attributes for terms. Hence, one must resort to ontologies if any of these features are required. A simple example of taxonomy is the Linnaean taxonomy of the living beings where the father-son relationship is represented by the type-of pair. In Fig 3.3 we can see an example of this taxonomy for human classification.



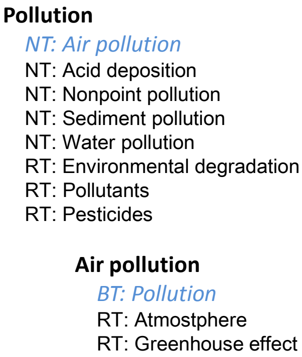
Domain: *Eukarya*
Kingdom: *Animalia*
Phylum: *Chordata*
Subphylum: *Vertebrata*
Class: *Mammalia*
Cohort: *Placentalia*
Order: *Primates*
Suborder: *Anthropoidea*
Infraorder: *Catarrhini*
Superfamily: *Hominoidae*
Family: *Hominidae*
Genus: *Homo*
Species: *Homo sapiens*

Fig 3.3 Linnaean taxonomy of the living beings: human classification

A **thesaurus** contains a set of relationships among concepts, organized in a taxonomic way. We may understand a thesaurus as a taxonomy together with a set of semantic relationships such as: equivalence, inverse, association, etc. The ANSI/ISO Monolingual Thesaurus standard defines the word thesaurus as: "*A controlled vocabulary arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardized relationship indicators that are employed reciprocally. The primary proposes of a thesaurus are: a) to facilitate retrieval of documents and b) to achieve consistency in the indexing of writing or otherwise recorded documents and other items, mainly for postcoordinate information storage and retrieval systems*". In a thesaurus, the set of allowed relationships that can hold between the concepts is finite and well defined. This set sometimes includes well known real-world relationships such as: part-of, member-group, stage-process, place-region, material-object, cause-effect, etc. If relationships other than those thesauri support are required, one must resort to more general ontologies. A well-known example of a the-

[20] http://delicious.com/
[21] http://www.flickr.com/

saurus is AGROVOC[22]. AGROVOC is a multilingual structured thesaurus of all subject fields in Agriculture, Forestry, Fisheries, Food security and related domains. It consists of words or expressions (terms), in different languages and organized in relationships (e.g., “broader”, “narrower”, and “related”), used to identify or search resources. It was developed by the Food and Agriculture Organization of the United Nations (FAO) and the Commission of the European Communities, in the early 1980s and it is updated roughly every three months. Fig 3.4 shows an example of the AGROVOC structure representing the terms Pollution and Air Pollution. For each term, a block is displayed, showing the hierarchical and non-hierarchical relations to other terms: BT (broader term), NT (narrower term), RT (related term), UF (non-descriptor).



Pollution
NT: Air pollution
NT: Acid deposition
NT: Nonpoint pollution
NT: Sediment pollution
NT: Water pollution
RT: Environmental degradation
RT: Pollutants
RT: Pesticides

Air pollution
BT: Pollution
RT: Atmostphere
RT: Greenhouse effect

Fig 3.4 Example of AGROVOC thesaurus structure: pollution and air pollution.

The most agreed definition of **ontology** was made by Gruber in 1993 (Gruber, A Translation Approach to Portable Ontology Specifications, 1993): “*An Ontology is a formal, explicit specification of a shared conceptualization*”. *Conceptualization* refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena. *Explicit* means that the type of concepts used and the constraints on their use are explicitly defined. *Formal* refers to the fact that the ontology should be machine-readable. *Shared* reflects that an ontology should capture consensual knowledge accepted by different communities. Going further in Gruber´s view, an ontology can be seen as the representation of knowledge in a domain, where objects and their relationships are described.

A more succinct definition comes from the W3C: “*Ontology defines the terms used to describe and represent an area of knowledge. It includes computer-usable definitions of basic concepts in a domain and the relationships among them*”. In this definition ontologies describe artifacts with different degrees of structure that specify descriptions for the following kinds of concepts: a) classes (general things) in the many domains of interest; b) relationships that can exist among things c) properties (or attributes) those things may have and d) restrictions or constraints impose to concepts.

[22] *www.fao.org/agrovoc/*

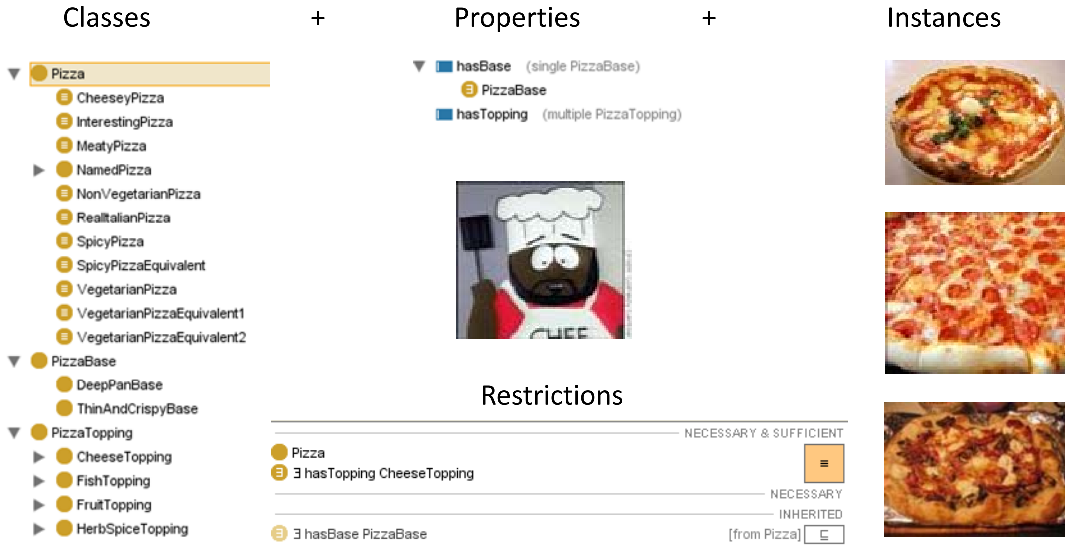


Fig 3.5 Graphical representation of an ontology using the Protégé[23] tool

Fig 3.5 is a graphical example of the ontology definition. The **ontology** is composed by *classes* that represent conceptual meanings about the pizza domain, *properties* and *restrictions* over these classes. In a second layer we have the **Knowledge Bases** (KBs) or groups of instances that represent the real individuals of the ontological concepts.

In practical terms, ontologies are commonly handled as hierarchies of concepts with attributes and relations, which establish a terminology to define semantic networks of interrelated concepts and instances, describing domain-specific knowledge which is stored in a KB. In many ways, an ontology is similar to a thesaurus. Fundamental and practical differences can be noted nonetheless. While a thesaurus usually has a pre-established set of relation types, ontologies tend to be more flexible, typically open to arbitrary relation types, which can be potentially extended anytime. The emphasis on formalization is much higher, which seeks to describe the world (or at least a domain) on the basis of a descriptive logic which axiomatizes the classes, their relations, and the properties in suitable terms to be formally reasoned upon. In this sense, it is generally considered that a thesaurus is a particular case of ontology, the latter bearing a considerably higher expressive power.

On the other hand, ontological KBs tend to be oriented (though not always) to storing large amounts of knowledge, with a much finer level of detail than is usually envisioned in a thesaurus. We might say that, in a way (leaving aside the variety of cases, which can be considerably wide) these KBs are conceived with an intermediate perspective between a database and a thesaurus. The potential of a resource of this kind is clear and proportional to its level of detail and coverage, as well as its development and maintenance cost, as it has been pointed out long since (Croft, 1986).

To conclude we can say that, taxonomies, thesaurus and ontologies are formal semantic knowledge representations that help to structure, classify, model, and or represent the concepts and rela-

[23] http://protege.stanford.edu/

tionships pertaining to some subject matter of interest to some community. The three of them are intended to enable a community to come to agreement and to commit to use the same terms in the same way. However, while taxonomies and thesauri may relate terms in a controlled vocabulary via parent-child and associative relationships, they do not contain explicit grammar rules to constrain how to use controlled vocabulary terms to express or model something meaningful within a domain of interest. In that sense, formalization is much higher in ontologies, making it easier for machines to understand the semantics behind services and contents. Therefore, ontologies are the main the knowledge representation model adopted in this thesis work.

3.2.1.1 Ontology classifications

As we stress in the previous section, ontology-based semantic technologies positively uphold the intense use of semantic knowledge with diverse purposes. Ontologies present different levels of detail and generality of the captured conceptualizations, which entails different ontology classifications. Among the most popular ones we can highlight: a) the one made by Guarino in 1998 (Guarino, Formal Ontology and Information Systems, 1998) considering the different generality of ontologies and b) the one made by Gómez Pérex in 2003 (Gómez-Pérez, Fernández-López, & Corcho, Ontological Engineering, 2003) considering the information represented by the ontology as the main classification criteria. Both of them are sown in Table 3.1.

Guarino classification (Guarino, Formal Ontology and Information Systems, 1998)	**Asunción Gómez Pérez** (Gómez-Pérez, Fernández-López, & Corcho, Ontological Engineering, 2003)
• *Upper Level Ontologies*: describe very general concepts like space, time, matter, object, event, action, etc., which are independent of a particular problem or domain and can be reused to construct new ontologies. • *Domain Ontologies*: describe the vocabulary related to a generic domain by specializing the concepts provided by the Upper-level ontology. • *Task Ontologies*: describe the vocabulary related to a generic task or activity by specializing the terms introduced in the Upper-level ontology. • *Application Ontologies:* describe concepts depending both on a particular domain and task, which are often specializations of both related ontologies. These concepts often correspond to roles played by domain entities while performing a certain activity.	• *Knowledge Representation ontologies*: provide primitive modeling elements of knowledge representation models. They offer modeling constructs used in frame-based representations, such as classes, subclasses, values, attributes and axioms. • *Generic and common use ontologies*: represent common-sense knowledge that can be used in different domains. They typically include a vocabulary that relates classes, events, time, space, causality and behaviour, among other concepts. • *Upper level ontologies*: describe general concepts. • *Domain ontologies*: offer concepts that can be reused within a specific domain (medical, pharmaceutical, law among others). • *Task ontologies*: describe the vocabulary related to a task or activity (goals, schedules, etc). • *Domain-task ontologies*: are task ontologies that can be reused in one specific domain, but not generically in similar domains. • *Method ontologies*: provide definitions for concepts and relationships relevant to a process.

	- *Application ontologies*: contain all the necessary concepts to model the application. This kind of ontology is used to specialize and extend domain or task ontologies for a specific application. Ontology Classification schemes.

Table 3.1 Ontology classification schemas

3.2.1.2 Ontology description Languages

In order to represent and manage semantic knowledge, ontology-based technologies include: **ontology description languages**, ontology parsers, ontology query languages, ontology development environments and ontology management modules among other tools and libraries.

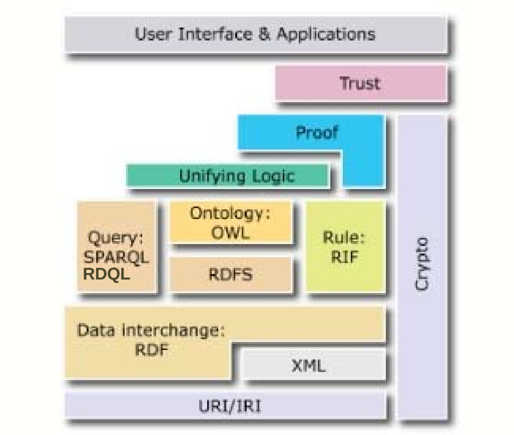


Fig 3.1 Ontology-based semantic technologies stack[24]

Ontology description languages provide the means to formally represent semantic knowledge. The layered model for ontology-based technologies, described in Fig 3.1, contains an illustration of the hierarchy of ontology description languages, where each layer exploits and uses capabilities of the layers below.

The **bottom layers** refer to the traditional hypertext Web languages and mechanisms. They constitute the pillars of ontology description languages, and therefore, the pillars for representing ontology-based semantic knowledge:

- Referencing mechanisms: Internationalized Resource Identifier (*IRI*), generalization of *URI*, provides means for uniquely identifying ontological resources.
- Document exchange standards:

[24] This figure is a modification of architecture vision for the semantic Web. (Berners-Lee, T. *Semantic Web- XML200*. Available at: http://www.w3.org/2008/Talks/0307-Tokyo-IH/HTML/img6.html)

  - *XML* is a markup language that enables creation documents of structured data. It provides a surface syntax for structured documents, but imposes no semantic constraints on the meaning of these documents.
  - *XML Namespaces* provides a way to use markups from different sources. They are used to refer to different sources in one document.
  - *XML Schema* is a language for restricting the structure of XML documents. It also extends XML with datatypes.

The **medium layers** refer to the current standardized ontology-based description and query languages. They constitute the formal way to represent and query ontology-based semantic knowledge:

- *Resource Description Framework (RDF)* is a language for creating a data model for objects (or "resources") and relations among them. It enables to represent information in the form of graph.
- *Resource Description Framework Schema (RDFS)* provides basic vocabulary for describing properties and classes of RDF resources. Using RDFS it is possible to create hierarchies of classes and properties.
- *Web Ontology Language (OWL)* extends RDFS by adding more advanced constructs to describe the semantics of RDF statements. It allows stating additional constraints, such as for example cardinality, restrictions of values, or characteristics of properties like transitivity. It appears as a way to capture more semantics and formally describe the meaning of terminology used in Web documents. It is based on *description logic* and so brings reasoning power to the knowledge representation.
- *RDF Data Query Language (RDQL)* and ***S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage (SPARQL)* are ontology query languages. They are used to extract specific information from RDF graphs.

The **final layers** contain ontology-based semantic technologies that are not yet standardized or ideas that should be implemented on the top of ontology-based semantic knowledge representations:

- *RIF* or *SWRL* will bring support of rules. This is important, for example, to allow describing relations that cannot be directly described using the OWL description logic.
- *Cryptography* is important to ensure and verify that RDF statements are coming from trusted sources. This can be achieved by appropriate digital signature of RDF statements.
- *Trust* to derived statements will be supported by (a) verifying that the premises come from trusted sources and by (b) relying on a formal logic for deriving new information.
- *User interface* is the final layer that will enable humans to use ontology-based semantic applications and therefore to exploit ontology-based semantic knowledge.

Ontology-based description and query languages play an important role in the type of knowledge representation used in this work. A brief section is added here to explain the main characteristics of each of these languages.

3.2.1.2.1 RDF and RDFS

The **Resource Description Framework Schema (RDFS)** is used to define hierarchies of classes, specifying their properties and the relations among them.

The **Resource Description Framework (RDF)** is a general-purpose language for representing information in the Web. It is particularly intended for representing information about Web resources. The RDF metadata model is based upon the idea of making statements about resources in the form of subject-predicate-object expressions, called *triples* in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object.

For example, Fig 3.2 shows one way to represent the notion "Starry Nigth was created by Van Gogh" using RDF and RDFS languages. While RDFS is used to represent the conceptual hierarchy of classes and relationships, RDF represents the information as a triple containing: a subject denoting "Starry Nigth", a predicate denoting "has author", and an object denoting "Van Gogh ".

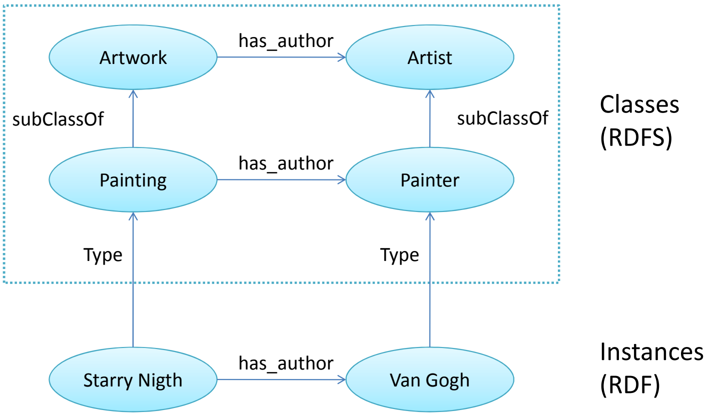
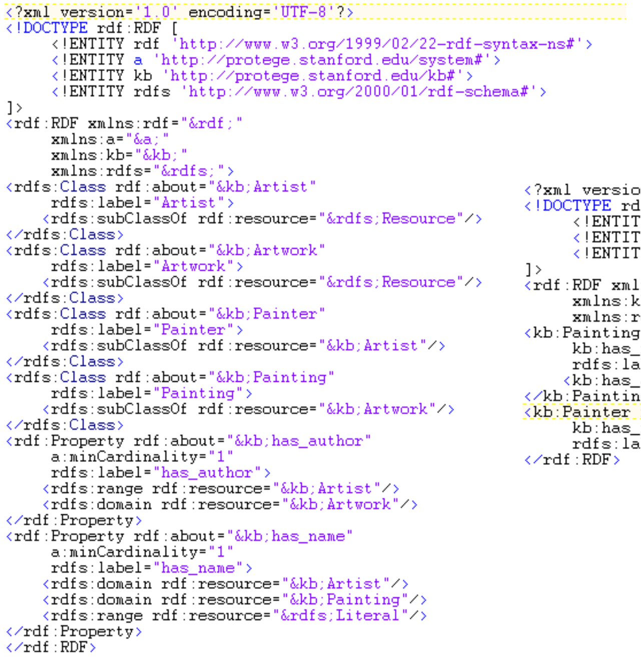


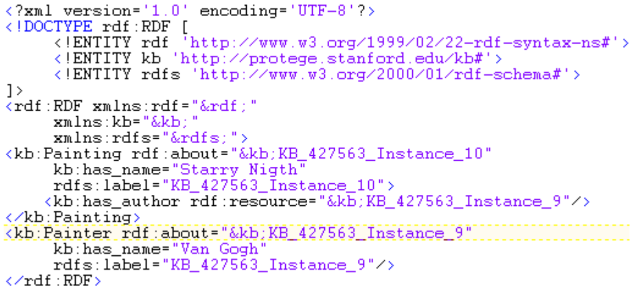
Fig 3.2 RDF and RDFS example

This conceptualization is translated into the ontology RDF and RDFS syntax described in Fig 3.3.



```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
     <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
     <!ENTITY a 'http://protege.stanford.edu/system#'>
     <!ENTITY kb 'http://protege.stanford.edu/kb#'>
     <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
     xmlns:a="&a;"
     xmlns:kb="&kb;"
     xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&kb;Artist"
     rdfs:label="Artist">
    <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&kb;Artwork"
     rdfs:label="Artwork">
    <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&kb;Painter"
     rdfs:label="Painter">
    <rdfs:subClassOf rdf:resource="&kb;Artist"/>
</rdfs:Class>
<rdfs:Class rdf:about="&kb;Painting"
     rdfs:label="Painting">
    <rdfs:subClassOf rdf:resource="&kb;Artwork"/>
</rdfs:Class>
<rdf:Property rdf:about="&kb;has_author"
     a:minCardinality="1"
     rdfs:label="has_author">
    <rdfs:range rdf:resource="&kb;Artist"/>
    <rdfs:domain rdf:resource="&kb;Artwork"/>
</rdf:Property>
<rdf:Property rdf:about="&kb;has_name"
     a:minCardinality="1"
     rdfs:label="has_name">
    <rdfs:domain rdf:resource="&kb;Artist"/>
    <rdfs:domain rdf:resource="&kb;Painting"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
</rdf:RDF>
```

RDFSchema



```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
     <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
     <!ENTITY kb 'http://protege.stanford.edu/kb#'>
     <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
     xmlns:kb="&kb;"
     xmlns:rdfs="&rdfs;">
<kb:Painting rdf:about="&kb;KB_427563_Instance_10"
     kb:has_name="Starry Nigth"
     rdfs:label="KB_427563_Instance_10">
    <kb:has_author rdf:resource="&kb;KB_427563_Instance_9"/>
</kb:Painting>
<kb:Painter rdf:about="&kb;KB_427563_Instance_9"
     kb:has_name="Van Gogh"
     rdfs:label="KB_427563_Instance_9"/>
</rdf:RDF>
```

RDF

Fig 3.3 RDF and RDFS syntax: generated using the Protégé tool

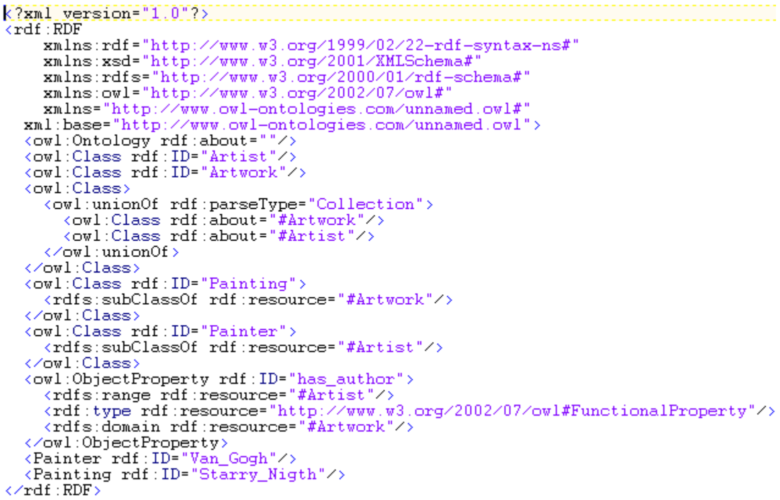
3.2.1.2.2 OWL

The OWL Ontology Language facilitates greater machine interpretability of Web content than the one supported by RDF and RDFS by providing additional vocabulary along with formal semantics like: relations between classes (e.g. disjointness), different type of properties (ObjectProperty, DatatypeProperty) characteristics of properties (e.g. symmetry), enumerated classes, etc.

OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

- *OWL Lite* supports a classification hierarchy and simple constraint features. E.g., while OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1.
- *OWL DL* supports maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL DL includes restrictions such as type separation (a class can not also be an individual or property; a property can not also be an individual or class).
- *OWL Full* supports maximum expressiveness and the syntactic freedom of RDF with no computational guarantees for reasoning. E.g., in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right.

Fig 3.4 represents in OWL the conceptualization described in Fig 3.2.



```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Artist"/>
  <owl:Class rdf:ID="Artwork"/>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Artwork"/>
      <owl:Class rdf:about="#Artist"/>
    </owl:unionOf>
  </owl:Class>
  <owl:Class rdf:ID="Painting">
    <rdfs:subClassOf rdf:resource="#Artwork"/>
  </owl:Class>
  <owl:Class rdf:ID="Painter">
    <rdfs:subClassOf rdf:resource="#Artist"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="has_author">
    <rdfs:range rdf:resource="#Artist"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:domain rdf:resource="#Artwork"/>
  </owl:ObjectProperty>
  <Painter rdf:ID="Van_Gogh"/>
  <Painting rdf:ID="Starry_Nigth"/>
</rdf:RDF>
```

Fig 3.4 OWL syntax: generated using the Protégé tool

3.2.1.2.3 RDQL and SPARQL

A pending achievement of ontology-based knowledge technologies for many years was the creation of query languages. Ontology query languages are used to express complex searches on the ontological graph through a simple declarative syntax.

Before the standardization of an official ontology query language, there were several de facto standards. One of the most popular ones was the RDF Query Language, known as RDQL. It was developed by Hewlett Packard laboratories. RDQL is a “data-oriented” query language in the sense that it only queries the information held in the ontologies; but is not able to provide any inference at query time. To perform a query, RDQL provides a way of specifying a graph pattern that is compared against the graph to yield a set of matches

In January 2008, the W3C organization announced SPARQL as the standard ontology query language. SPARQL presents several advantages in comparison to RDQL, like the possibility to create RDF models out of a query.

```
PREFIX kb:   <http://protege.stanford.edu/kb#>
SELECT ?painting ?painter
WHERE
  { ?painting kb:has_author ?painter }
```

Fig 3.5 SPARQL example

Fig 3.5 presents a small example of a SPARQL query over the ontology described in Fig 3.2. The query asks the ontology for all the possible paintings and their corresponding authors, where *“?painting”* represents all the possible instances of painting within the ontology, and *“?painter”* represents all their possible authors.

3.3 Semantic knowledge acquisition

Acquiring semantic knowledge is, in general, a high-cost process that requires human intervention. In the case of ontologies, the process is particularly expensive, requiring the service of experts both in ontology engineering and the domain of interest. While this may be acceptable in some high value applications, for the widespread adoption of ontology-based semantic knowledge, some sort of semi-automatic approaches to ontology-based knowledge acquisition is required.

The proposal of this thesis inherits all the well-known problems of ontology-based knowledge acquisition, from building and sharing well-defined ontologies to populating KBs. In the last few years, several initiatives have aimed to solve this problem, like the OLP (Ontology Learning and Population) annual workshop promoted by the ECAI (European Conference on Artificial Intelligence), resulting in significant contributions to the literature (Cimiano P. , 2006).

The following sections aim to briefly introduce the problems of semi-automatic ontology learning and population and point some relevant works in the area.

3.3.1 Ontology construction

Most of the achievements towards semi-automatic ontology construction address one of the following problems/tasks:

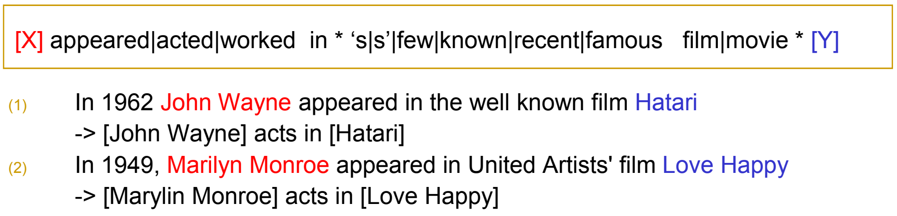
- *Extending a given ontology*: Given an ontology with concepts and relations, the goal is to extend that ontology using a text corpus. E.g., Aguirre's work (Aguirre, Ansa, Hovy, & Martínez, 2000) uses Web documents to extend large ontologies.
- *Learning relations for a given ontology*: Given a collection of text documents and an ontology with concepts, the goal is to learn relations between the concepts. The approaches include learning taxonomic, for example "is-a relations" (Cimiano, Pivk, Schimidt-Thieme, & Staab, 2004) as well as non-taxonomic, for example "has-part" relations (Maedche & Staab, Discovering Conceptual Relations from text., 2000). They also include extracting semantic relations from text based on collocations (Heyer, Laüter, Quasthoff, Wittig, & Wolff, 2001).
- *Ontology construction based on clustering*: Given a collection of text documents, the goal is to: split each document into sentences, parse the text and apply clustering for semi-automatic construction of an ontology (Bisson, Nédellec, & Cañamero, 2000). Each cluster is labelled by the most characteristic words from its sentences.
- *Ontology construction based on named entities*: Given a collection of news, the goal is to represent it as a collection of graphs, where the nodes are named entities extracted from the text and relationships between them are based on the context and collocation of the named entities (Grobelnik & Mladenic, 2004).

3.3.2 Ontology population

Ontology population deals with the task of identifying new instances belonging to concepts in a given ontology and enriches KBs using the identified instances and their semantic relationships. Similarly to ontology construction, ontology population methodologies are high cost processes that require major engineering efforts. Attempting to solve this problem, automatic ontology population from texts has emerged as a field of application for ontology-based knowledge acquisition techniques.

There are two main paradigms distinguishing ontology population approaches:

- *Pattern-based approaches*: Pattern-based approaches search for phrases which explicitly show that there is a relation between two words (Hearst, 1992) (Ruiz, Alfonseca, & Castells, 2007). Fig 3.6 shows an example of how to apply a pattern to extract the relation actor-film. The name of the actor is represented by "X", and the film title is represented by "Y". In between, the pattern considers potential verbs related to an actor performance, like "appearing or acting or working" in a movie. It also considers adjectives that express the quality of movies: "famous, recent, known, few…" Finally, this pattern has the word "film or movie" until it gets the name of the film.



[X] appeared|acted|worked in * 's|s'|few|known|recent|famous film|movie * [Y]

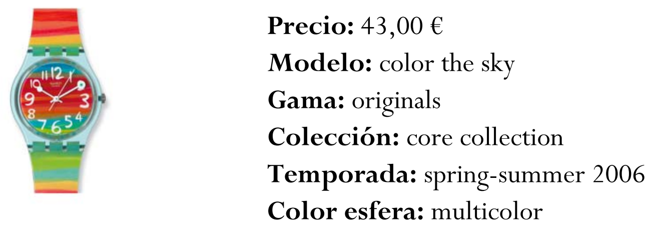
(1) In 1962 John Wayne appeared in the well known film Hatari
-> [John Wayne] acts in [Hatari]

(2) In 1949, Marilyn Monroe appeared in United Artists' film Love Happy
-> [Marylin Monroe] acts in [Love Happy]

Fig 3.6 Example of a pattern to extract the relation actor-film[25]

- *Wrapper-based approaches*: Wrapper-based approaches analyze a corpus to extract features from the context in which a semantic class tends to appear (Crescenzi & Mecca, 2004). Fig 3.7 shows an example of a wrapper-based approach that, departing from a site of watches analyzes the Web pages structure and extracts relevant data to populate ontologies with instances of watches and properties such as their price, model, collection, etc.

[25] Extracted from (Ruiz, Alfonseca, & Castells, 2007)



Precio: 43,00 €
Modelo: color the sky
Gama: originals
Colección: core collection
Temporada: spring-summer 2006
Color esfera: multicolor

Fig 3.7 Example of a Website to extract information about watches

State-of-the-art approaches may also be divided into two classes, according to different use of training data: *Unsupervised approaches* and *supervised approaches*. While state of the art unsupervised methods have low performance, supervised approaches reach higher accuracy, but require the manual construction of a training set, which impedes them from large-scale applications.

3.4 Semantic knowledge annotation

All the meanings and information conveyed by content in unstructured form (such as text or audiovisual content) cannot in general be fully translated to a clear and formal semantic representation, for both pragmatic (cost) and intrinsic (problems for the formalization of the world) reasons. However, it is possible to formally describe parts of the conveyed information, albeit to an incomplete extent, as metadata. Metadata is data about other data (e.g., the ISBN number and the author's name are metadata about a book). For the same reason that it is generally useful to keep both parts of information (data and metadata) in the system, it is also relevant to have a link that connects the two of them, commonly known as **annotation**.

Different syntactic supports and standards have been proposed for the representation of metadata and annotations. Markup languages like HTML and XML are widespread nowadays, but they have limitations in their expressiveness and share ability (Passin, 2004). Ontology-based technologies have been developed in the last few years to address and overcome these limitations. For example, imagine a document that contains the keyword "jaguar". This keyword is ambiguous because it might refer to the animal or to the car. An ontology-based annotation can relate the word "jaguar", appearing in the document, to an ontology concept that defines "jaguar" as the abstract concept "animal", thus removing any ambiguity.

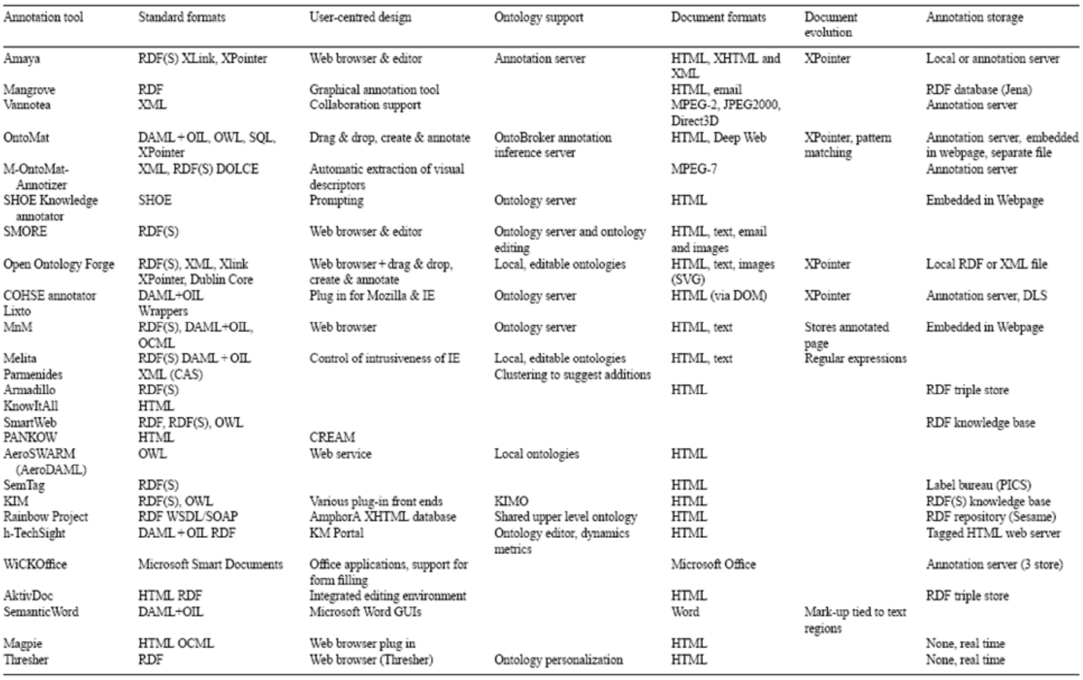
A survey of ontology-based technologies for semantic annotation is reported in (Uren, et al., 2006). This work proposes a document centric model for ontology-based semantic annotation that manages three elements: ontologies (metadata), documents (data, or content in unstructured form) and annotations (links between the data and the metadata).

They identify seven requirements for ontology-based semantic annotation systems:

- *Standard formats*: using standard formats is preferred whenever possible because the investment in making up resources is considerable and standardization builds in future proofing because new tools, services, etc.

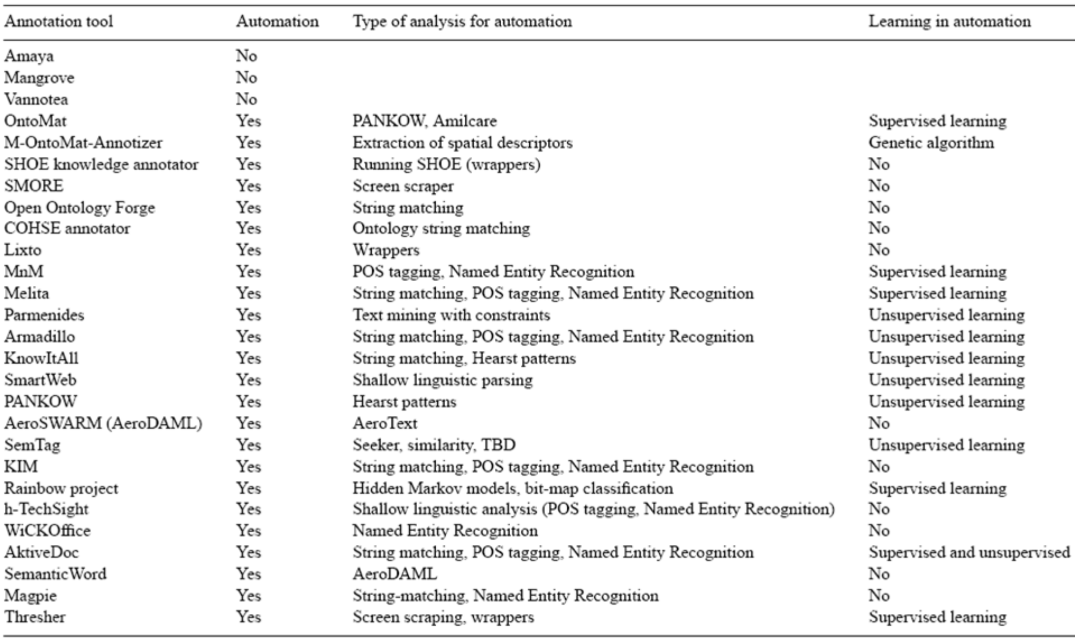
- *User centered/collaborative design*: in the case of manual annotation tools, it is crucial to provide users with easy to use interfaces that simplify the annotation process and place it in the context of their every day work.
- *Multiple Ontology support*: annotation tools need to be able to support multiple ontologies. For example, in a medical context, there may be one ontology for general metadata about a patient and other technical ontologies that deal with diagnosis and treatment.
- *Support of heterogeneous document formats*: standards for annotation tend to assume that the documents being annotated are in Web-native formats such as HTML and XML. However, with the emergence of new multimedia content in the Web, documents will be in many different formats (audio, video, etc).
- *Document evolution*: Ontologies and documents change continuously, which means that the annotation process should not be fixed.
- *Annotation storage*: The ontology-based semantic annotation model assumes that annotations will be stored separately from the original documents. However, many tools store the annotations as integral part of the documents and therefore they do not decouple data and metadata.
- *Automation*: an important aspect of easing the knowledge acquisition bottleneck is the provision of facilities for automatic mark up of document collections. To achieve this, the integration of knowledge acquisition technologies into the annotation environment is vital.

The work in (Uren, et al., 2006) also analyzes different annotation tools considering this seven annotation requirements. Fig 3.8 shows a comparison considering the first six requirements while Fig 3.9 represents just the automation requirement. As we can see in Fig 3.9, many systems have some kind of automatic and semi-automatic support for annotations.



Annotation tool	Standard formats	User-centred design	Ontology support	Document formats	Document evolution	Annotation storage
Amaya	RDF(S) XLink, XPointer	Web browser & editor	Annotation server	HTML, XHTML and XML	XPointer	Local or annotation server
Mangrove	RDF	Graphical annotation tool		HTML, email		RDF database (Jena)
Vannotea	XML	Collaboration support		MPEG-2, JPEG2000, Direct3D		Annotation server
OntoMat	DAML + OIL, OWL, SQL, XPointer	Drag & drop, create & annotate	OntoBroker annotation inference server	HTML, Deep Web	XPointer, pattern matching	Annotation server, embedded in webpage, separate file
M-OntoMat-Annotizer	XML, RDF(S) DOLCE	Automatic extraction of visual descriptors		MPEG-7		Annotation server
SHOE Knowledge annotator	SHOE	Prompting	Ontology server	HTML		Embedded in Webpage
SMORE	RDF(S)	Web browser & editor	Ontology server and ontology editing	HTML, text, email and images		
Open Ontology Forge	RDF(S), XML, Xlink XPointer, Dublin Core	Web browser + drag & drop, create & annotate	Local, editable ontologies	HTML, text, images (SVG)	XPointer	Local RDF or XML file
COHSE annotator	DAML+OIL	Plug in for Mozilla & IE	Ontology server	HTML (via DOM)	XPointer	Annotation server, DLS
Lixto	Wrappers					
MnM	RDF(S), DAML+OIL, OCML	Web browser	Ontology server	HTML, text	Stores annotated page	Embedded in Webpage
Melita	RDF(S) DAML + OIL	Control of intrusiveness of IE	Local, editable ontologies	HTML, text	Regular expressions	
Parmenides	XML (CAS)		Clustering to suggest additions			
Armadillo	RDF(S)			HTML		RDF triple store
KnowItAll	HTML					
SmartWeb	RDF, RDF(S), OWL					RDF knowledge base
PANKOW	HTML	CREAM				
AeroSWARM (AeroDAML)	OWL	Web service	Local ontologies	HTML		
SemTag	RDF(S)			HTML		Label bureau (PICS)
KIM	RDF(S), OWL	Various plug-in front ends	KIMO	HTML		RDF(S) knowledge base
Rainbow Project	RDF WSDL/SOAP	AmphorA XHTML database	Shared upper level ontology	HTML		RDF repository (Sesame)
h-TechSight	DAML + OIL RDF	KM Portal	Ontology editor, dynamics metrics	HTML		Tagged HTML web server
WiCKOffice	Microsoft Smart Documents	Office applications, support for form filling		Microsoft Office		Annotation server (3 store)
AktivDoc	HTML RDF	Integrated editing environment		HTML		RDF triple store
SemanticWord	DAML+OIL	Microsoft Word GUIs		Word	Mark-up tied to text regions	
Magpie	HTML OCML	Web browser plug in		HTML		None, real time
Thresher	RDF	Web browser (Thresher)	Ontology personalization	HTML		None, real time

Fig 3.8 Comparison of annotation tools considering the first six requirements[26]



Annotation tool	Automation	Type of analysis for automation	Learning in automation
Amaya	No		
Mangrove	No		
Vannotea	No		
OntoMat	Yes	PANKOW, Amilcare	Supervised learning
M-OntoMat-Annotizer	Yes	Extraction of spatial descriptors	Genetic algorithm
SHOE knowledge annotator	Yes	Running SHOE (wrappers)	No
SMORE	Yes	Screen scraper	No
Open Ontology Forge	Yes	String matching	No
COHSE annotator	Yes	Ontology string matching	No
Lixto	Yes	Wrappers	No
MnM	Yes	POS tagging, Named Entity Recognition	Supervised learning
Melita	Yes	String matching, POS tagging, Named Entity Recognition	Supervised learning
Parmenides	Yes	Text mining with constraints	Unsupervised learning
Armadillo	Yes	String matching, POS tagging, Named Entity Recognition	Unsupervised learning
KnowItAll	Yes	String matching, Hearst patterns	Unsupervised learning
SmartWeb	Yes	Shallow linguistic parsing	Unsupervised learning
PANKOW	Yes	Hearst patterns	Unsupervised learning
AeroSWARM (AeroDAML)	Yes	AeroText	No
SemTag	Yes	Seeker, similarity, TBD	Unsupervised learning
KIM	Yes	String matching, POS tagging, Named Entity Recognition	No
Rainbow project	Yes	Hidden Markov models, bit-map classification	Supervised learning
h-TechSight	Yes	Shallow linguistic analysis (POS tagging, Named Entity Recognition)	No
WiCKOffice	Yes	Named Entity Recognition	No
AktiveDoc	Yes	String matching, POS tagging, Named Entity Recognition	Supervised and unsupervised
SemanticWord	Yes	AeroDAML	No
Magpie	Yes	String-matching, Named Entity Recognition	No
Thresher	Yes	Screen scraping, wrappers	Supervised learning

Fig 3.9 Comparison of annotation tools considering the automation requirement[27]

[26] Extracted from (Uren, et al., 2006)

[27] Extracted from (Uren, et al., 2006)

3.5 Semantic knowledge evaluation

What is high-quality semantic knowledge? This question is crucial from a practical perspective, when we think about the development of semantic-based knowledge technologies.

Different methodologies for ontology-based semantic knowledge evaluation have been proposed in the literature considering the characteristics of the ontologies and the specific goals or tasks that the ontologies are intended for. An overview of ontology evaluation approaches is presented in (Brank, Grobelnik, & Mladenić, 2005), where four different categories are identified:

- Those that evaluate an ontology by comparing it to a Golden Standard, which may itself be an ontology (Maedche & Staab, 2002) or some other kind of representation of the problem domain for which an appropriate ontology is needed.
- Those that evaluate the ontologies by plugging them in an application, and measuring the quality of the results that the application returns (Porzel & Malaka, 2004).
- Those that evaluate ontologies by comparing them to unstructured or informal data (e.g., text documents (Brewester, 2004)) which represent the problem domain.
- Those based on human interaction to measure ontology features not recognizable by machines (Lozano-Tello & Gómez-Pérez, 2004).

In each of the above approaches, a number of different evaluation levels might be considered to provide as much information as possible. Several levels can be identified in the literature:

- The lexical level (Brewester, 2004) (Maedche & Staab, 2002) (Velardi, Navigli, Cuchiarelli, & Neri, 2005) which measures the quality by comparing the words (lexical entries) of the ontology with a set of words that represent the problem domain.
- The taxonomy level (Maedche & Staab, 2002) which considers the hierarchical connection between concepts using the *is-a* relation.
- Other semantic relations besides hierarchical ones (Gangemi, Catenacci, Ciaramita, & Lehmann, 2005) (Guarino & Welty, 2002).
- The syntactic level which considers the syntactic requirements of the formal language used to describe the ontology (Gómez-Pérez, 1995).
- Context or application level (Ding, Finin, Joshi, Pan, & Cost, 2004) which considers the context of the ontology, such as the ontologies that reference or are referenced by the one being evaluated, or the application it is intended for.
- The structure, architecture and design levels (Lozano-Tello & Gómez-Pérez, 2004) which take into account the principles and criteria involved in the ontology construction itself.

	Golden Standard	Application based	Data Driven	Assessment by humans
Lexical entries, vocabulary, concept data	X	X	X	X
Hierarchy, taxonomy	X	X	X	X
Other semantic relations	X	X	X	X
Context, application		X		X
Syntactic	X			X
Structure, architecture, design				X

Table 3.2 An overview of approaches to ontology evaluation[28]

[28] Extracted from (Brank, Grobelnik, & Mladenić, 2005)

3.6 Summary

In this chapter, we survey the development towards semantic-intensive knowledge technologies in the last decade with the aim to automate tasks using software that substitutes human knowledge. We introduce and revise the different problems for semantic-based knowledge representation, acquisition, annotation and evaluation.

Semantic knowledge representation is the study of how knowledge about the world can be represented and what kinds of reasoning can be done with that knowledge. Important questions in semantic knowledge representation include the tradeoffs between representational adequacy, fidelity, acquisition cost and computational cost. Considering these tradeoffs, we identify four different semantic knowledge representations: *Bags of words* (uncategorized terms), *Taxonomies* (categories + hierarchical relations), *Thesauri (*categories + fixed hierarchical and associative relations) and, *Ontologies* (classes + instances + arbitrary semantic relations + rules). Because of its level of formalization, ontologies are presented as the main knowledge representation adopted in this thesis work. With the aim to further study the technologies to represent and query the information stored in the ontologies we present several ontology description languages (RDF, RDFS and OWL) and ontology query-languages (RDQL and SPARQL).

Semantic knowledge acquisition is the study of how knowledge about the world can be obtained. This is in general, a high-cost process that requires human intervention. In the case of ontologies, the process is particularly expensive, requiring the service of experts both in ontology engineering and the domain of interest. While this may be acceptable in some high value applications, for the widespread adoption of ontology-based semantic knowledge, some sort of semi-automatic approaches to ontology-based knowledge acquisition is required. In this chapter we briefly introduce the problems of semi-automatic ontology learning and population and point some relevant works in the area.

Semantic knowledge annotation is the study of how the meanings and information conveyed by content in unstructured form (such as text or audiovisual content) can be formally described, albeit to an incomplete extent, with metadata. We introduce the survey of ontology-based technologies for semantic annotation reported in (Uren, et al., 2006), which proposes a document centric model for ontology-based semantic annotation that manages three main elements: ontologies (metadata), documents (data, or content in unstructured form) and annotations (links between the data and the metadata).

Semantic Knowledge evaluation is the study of techniques and measures that determine what “high quality semantic knowledge” is. We introduce the survey of ontology-based evaluation methodologies presented in (Brank, Grobelnik, & Mladenić, 2005), which proposes four different categories (golden standard, application-based, data driven and assessment by humans) and six evaluation levels (lexical, taxonomical, other semantic relations, context, syntactic, structure) to determine the quality of ontologies.

Chapter 4

Related work

This chapter revises the notion of **semantic search** (section 4.1). It seeks a comprehensive perspective by revising the work in IR field since the early days (section 4.2) up to the latest prospects arisen from the Semantic Web area (section 4.3). It also presents an initial classification of the studied systems according to their different knowledge representations (section 4.4). A final overview is shown to provide a panoramic view of the area, in order to identify key dimensions in the formulation of the problem, past and current achievements, difficulties and potential directions ahead (section 4.5).

4.1 What is semantic search?

Any IR system is based on a logic representation of user information needs, and the information supplied by the information objects in the search space, in such a way that the comparison between queries and potential answers takes place in the ideal model (section 2.2). The various logic representations proposed in the area (Lewis & Gale, 1994) respond, on the one hand, to the requirement of being efficiently processable by an IR system, and necessarily entail some information loss. This is clear, for instance, in the representation of information needs by a simple list of keywords, as is the case in currently dominant technology in both research and industry.

On the other hand, an underlying goal to any IR system is that the observations performed in the ideal model correlate as frequently as possible with equivalent observations by real users. In this aim, it is natural to consider the idea of reducing the distance between the logic representation in the system and the real one in the user's mind, with regards to the formulation of queries and the understanding of documents. The problem is complex, not just by the contraposition between this ideal and the automation requirement, but also due to the involvement of diverse, difficult to capture, aspects related to human cognition, and even the definition of reality, truth, and meaning.

Among other reasons, this can account for the fact that the widely adopted representation in the IR field is the so-called bag of words (for text content), by which the comparison between queries and answers is mainly based on literal coincidences between queries and document passages. Similarly, syntactic or numeric features semantically "dry" and close to the pixel level, are used for visual content.

Nonetheless, many efforts have explored the possibility to elaborate the representational level beyond the literality of character strings or signal features, towards more abstract models that approximate a conceptual representation of sought and available information, in order to enhance the response accuracy and coverage for certain types of queries. We are also assisting to a renewed interest today in the search engine market towards the introduction of semantic capabilities in current search engines (Taylor, 2007) (see e.g. Hakia[29], Powerset[30], AskMeNow[31] and Digger[32], to name a few).

In this chapter we revise the issue of using semantic and domain knowledge in IR, seeking a comprehensive perspective, revisiting the related work undertaken in the IR field since the early eighties (or earlier) up to the latest efforts, and the related prospects arisen from semantic-based technologies, as a younger area specifically **focusing on the issues of domain semantics representation**. Our study is motivated by basic underlying questions such as what doing semantic search means, what has been achieved, where we are standing, what further progress is possible, and in which directions.

[29] http://www.hakia.com
[30] http://www.powerset.com
[31] http://www.askmenow.com
[32] https://www.digger.com

4.2 Semantic search: an IR perspective

The elaboration of conceptual frameworks and their introduction in IR models has wide precedents. The following quotation from a work by W. B. Croft published more than twenty years ago (Croft, 1986) could well serve today as an introduction to the topic at hand:

> "*The systems that have been developed, such as those based on probabilistic models of relevance* (Van Rijsbergen, 1979)*, capture 'domain knowledge' purely in the statistics of occurrence of individual words (or stems) in the documents and in statistical dependencies that exist between words. We define domain knowledge to mean information about the important topics or concepts in a particular domain and how they relate to each other. The statistical approach has many advantages and can achieve a reasonable level of effectiveness with techniques that are very efficient. However, it appears that to achieve significant improvements in retrieval effectiveness compared to current techniques, systems must be designed to acquire and use explicit domain knowledge.*"

Starting from this point of view, in the representation proposed by Croft the domain is modeled as a **thesaurus** of concepts, each one of which has a name, relations to other concepts, and a list of more or less ad-hoc rules (defined on a per-case basis) to recognize the concept in a textual passage. The considered relations between concepts include synonymy, hyponymy and instantiation, meronymy and similarity. This semantic knowledge is used to expand both queries and the document indexing entries through the relations between concepts. Aware of the cost of producing domain knowledge, Croft suggests using such knowledge as an enabler of incremental improvement over purely statistic methods, in such a way that the performance of the latter is retained in the absence or incompleteness of the former. Moreover, and to further address the incompleteness problem, Croft proposes the acquisition of domain knowledge by means of dialogs with the user, which can be seen as a far precedent of current proposals in the area of folksonomies (Gruber, 2008).

Croft's work is representative of a trend which, by that same period, explores the enhancement of IR systems' performance through the enrichment of the representation of meanings by introducing an explicit conceptual abstraction. In this line, and possibly under the influence of knowledge based systems in the Artificial Intelligence field, works proliferate in the eighties which investigate the use of **semantic networks** to enrich the representation of the indexing terms. See for instance: (Cohen & Kjeldsen, 1987) (Rau, 1987) (Shoval, 1981). The introduction of a conceptual model of this kind is motivated and developed in an even more explicit way in later works, such as the ones by Agosti and Crestani (Agosti, Crestani, Gradenigo, & Mattiello, 1990) (Agosti, Melucci, & Crestani, 1995) (Crestani, 1997) in which semantic relations are used in relevance propagation and assisted navigation strategies, in addition to query formulation. It is also interesting, and seminal of posterior works, the explicit distinction in the latter works of three representational levels (documents, words, and concepts), with relations within and between such levels.

The idea of augmenting the semantic representation of a document beyond a set of plain words is in fact present in earlier works to those decades, such as Karen Spärck Jones' PhD thesis itself as early as 1964 (Spärck Jones K. , 1964). In it, the author reflects on the flexible, non univocal correspondence between words and meanings, and the role of relations between words (synonymy, antonymy, hyponymy, entailment, and others) in the description of meanings. Her work considers the notion of predefined semantic primitives, consisting in essence of (domain-specific or general) concepts taken from a thesaurus (the Roget's), which are automatically extended with emergent semantic entities, observable in the analysis of a text corpus.

This work raises also the interesting question of whether or not "artificial" semantic resources should be used, external to what is strictly observable in the language use. This is a key question when it comes to using domain knowledge in IR, regarding which different authors, or the same ones at different times, have taken different stances. Spärck Jones' position in her thesis and trajectory thereafter may be defined by the principle of "*taking words as they stand*", put forward by herself (Spärck Jones K. , 2003). This position does not seem strict or absolute nonetheless, considering the use in her research of an "artificial" external resource such as the Roget's Thesaurus, constructed after the intuitive criteria of a single author. Although it could be argued that such thesaurus constitutes itself an observable linguistic phenomenon (Wilks & Tait, 2005), in such a way that the former premise was not contradicted, such consideration would but evidence the relative quality of the premise itself, when it comes to drawing a clear line between the observable and the artificial.

Works like the one by Croft, with which we started this section, lay many of the ideas and observations which in essence underlie subsequent work in that direction, but they can by no means be said to be anything near to a full development at that point. Considerable research followed indeed, in which several authors have kept progressing on conceptual approaches to IR based on domain knowledge, seeking a fuller development, an improvement of results, or their application to different scenarios (the Web, etc.) with their own characteristics and problems (scale, heterogeneity levels, user typology, etc.), addressing pending or new difficulties, and exploring the new opportunities brought by the evolution of technology.

One of the pursued lines in this direction is the one based on **linguistic approaches**, among which the use of resources like WordNet is particularly representative of the use of explicit conceptual descriptions (Madala, Takenobu, & Hozumi, 1998) (Vorhees, 1994). Although WordNet is a resource with domain-independence leaning, it can be said that in a way it captures knowledge, albeit generic or superficial, of a wide variety of domains.

Beyond WordNet, or complementarily to its use, many works have researched the use of thesauri with a higher or lower specialization level, to introduce enhancements in search effectiveness (Harbourt, Syed, Hole, & Kingsland, 1993) (Hersh & Greenes, 1990) (Hersh, Hickam, & Leone, 1992) (Järvelin, Kekäläinen, & Niemi, 2001) (Jones, 1993) (Paice, 1991) (Sanderson, 1994) (Yang & Chute, 1993). A thesaurus consists of a set of terms (words or titles) plus an arbitrary set of binary relations of different kinds (hierarchic, association, etc.), defined over the set of terms. In IR, thesauri represent an approximation (in rigor informal or intuitive, although their construction can be based on well-founded criteria and methodologies (American National Standards Institute, 1980)) to the representation of conceptual spaces, where the thesaural terms approximate concepts of the domain for which the thesaurus is built. One of the most common uses of thesauri in this context is the expansion of query terms, based on the mapping of query words to thesauri elements, and the extension of the latter through their relations to other terms in the thesaurus. It is common to use weights associated to the relations in the expansion, where the weights represent degrees of intensity in the relations, under different interpretations (certainty, similarity, etc.) and obtention methods (manual, statistic correlation, position in concept graphs, etc.) for such weights.

Both the use of manually created thesauri, and the automatic generation of the latter have been researched in the IR field. In the first case, they are usually built by domain experts in the subject where the thesaurus belongs. There is a multitude of specialized thesauri nowadays for the access to information in fields such as health, law, economy, arts, cultural heritage, education, different scientific areas, etc., which have been used in diverse works in this line. Given the cost involved in the construction and maintenance of a thesaurus, and the importance of the unified use of this type of resource, it is usual that thesauri undergo consensus and standardization for shared use. On its side, the automatic creation or extension of thesauri is generally based on the statistic analysis of the co-occurrence of thesaurus terms in passages from a text corpus, based on which relations between terms are inferred (Chen & Lynch, 1992) (Crouch, 1990).

The studies on the effectiveness of using thesauri yield uneven results, which too much extent depend on aspects such as the quality and degree of automation of the thesaurus construction, the use or not of relevance judgments provided by experts or users, the proximity between the corpus from which a thesaurus is generated, and the final search environment where it is applied, and other details such as the thesaurus term spotting techniques in text fragments. Although results have not been favorable in all cases (Hersh, Hickam, & Leone, 1992), there seems to be evidence or even consensus that it is possible to achieve improvements at least in relative terms (in some aspects, under certain conditions, etc.) by the use of thesauri (Yang & Chute, 1993).

The understanding of contents to retrieve more accurate answers is also long-term goal pursued by **Question Answering** (QA) approaches. The purpose of a QA system is to return answers, rather than documents containing answers, in response to a natural language query. IR-based QA approaches are classified as open QA approaches over free text, in contraposition to other QA approaches applied over semi-structured sources (like yellow pages), structures sources (like databases), or highly formalized sources (like ontologies). Open QA approaches over free text systems typically include a question classifier module that determines the type of question and the type of answer. After the question is analysed, the system uses several modules that apply increasingly complex NLP techniques on a gradually reduced amount of text. Finally, an answer extraction module looks for further clues in the text to determine if the candidate can indeed answer the question. In order to identify the type of question, various systems have built hierarchies of question types based on the types of answers sought (Hovy, Gerber, Hermjakob, Junk, & Lin, 2000) (Moldovan, et al., 1999) (Srihari, Li, & Li, 2004). For instance, in LASSO (Moldovan, et al., 1999) a question type hierarchy was constructed from the analysis of the TREC-8 training data. Question's classes are arranged hierarchically in taxonomies and different types of questions require different strategies. Some approaches have also exploited linguistic resources like WordNet to identify types of answers. For example, FALCON (Harabagiu, et al., 2000) identifies the expected answer type of the question "what do penguins eat?" as food because "it is the most widely used concept in the glosses of the sub-hierarchy of the noun synset {eating, feeding}". The integration of QA approaches can be seen nowadays in Web-scale commercial applications like the popular Ask (Ask.com) search engine.

From a very different starting point, the idea to raise IR techniques to a higher conceptual level is also explicitly present in **Latent Semantic Analysis** (LSA) techniques, widely studied and applied in diverse domains (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). Differently from thesauri-oriented techniques, concepts emerge in LSA by means of algebraic methods, based on the frequency of words in documents of a corpus. The method has the considerable advantage of not requiring the introduction of external knowledge to the corpus whatsoever. On the other hand, the resulting concepts from LSA are intangible in that they do not have any textual or intuitive expression of their own, but they are defined by vectors that relate them to words of the initial vocabulary. Concepts are thus mathematical abstractions here, which manifest themselves in the effect obtained from them when comparing queries and documents, documents between them, or words to other words. Related to this, and through such manifestations, LSA researchers have investigated the potential similarity between the pseudo-concepts found by LSA and the corresponding linguistic or cognitive phenomena, observable for instance in the detection of synonymy and antonymy relations, text classification, etc., by a person (Landauer, Foltz, & Laham, 1998). Although some correlation has been observed between the semantic associations obtained by LSA and human comprehension of meanings, the results in this realm cannot be considered conclusive, which limits the applicability of the product of LSA by itself to other contexts, as an explicit, reusable semantic resource or representation. Evidence has nonetheless been provided on the potential of this technique in terms of performance improvements in IR tasks (Dumais, 1994) (Letsche & Berry, 1997).

4.3 Semantic search: a SW perspective

The introduction of **ontologies** to move beyond the capabilities of current search technologies has been an often portrayed scenario in the area of semantic-based technologies since the late nineties (Luke, Spector, & Rager, 1996).

From the standpoint of an IR researcher, ontologies are commonly handled as hierarchies of concepts with attributes and relations, which establish a terminology to define semantic networks of interrelated concepts and instances, describing domain-specific knowledge which is stored in a knowledge base (KB). Compared to what is usual in thesauri, the emphasis on formalization is much higher in ontologies, which seek to describe the world (or at least a domain) on the basis of a descriptive logic which axiomatizes the classes, their relations, and the properties of both (symmetry, transitivity, equivalences, etc.), in suitable terms to be formally reasoned upon.

The most common way in which semantic search has been understood and addressed from the area of semantic-oriented technologies, especially in their beginnings in the late nineties, consists of the construction of a query engine that receives requests in an ontology query language (such as SPARQL today), executes them on a KB, and returns tuples of ontology values from the ontology which satisfy the conditions in the query. These techniques use thus Boolean search models, based on an ideal vision of the information space, as consisting of formal ontological knowledge units, devoid of ambiguity or redundancy. Under such perspective, the IR problem is reduced to a data retrieval task. A knowledge unit is an either correct or incorrect answer to a given information request, whereby the search results are assumed to be 100% precise, and there is no notion of approximate answer to an information need. This view can be framed as an issue of Question Answering (QA), a long researched topic in Natural Language Processing (Burger, 2001), also converging to the IR field (Vorhees, 2001).

The so-called semantic portals (Castells, Foncillas, Lara, Rico, & Alonso, 2004) (Contreras, et al., 2004) (Maedche, Staab, Stojanovic, Studer, & Sure, 2003) and ontology-based QA approaches (Lopez, Pasin, & Motta, 2005) (Lopez, Motta, & Uren, 2006) (Bernstein & Kaufmann, 2006) (Cimiano, Haase, & Heizmann, 2007) are a good example of this approach. These approaches typically provide simple search functionalities which may be better classed in the spectrum of semantic data retrieval, rather than semantic information retrieval. Searches return ontology instances or values, rather than documents, and no ranking method is usually provided. In some systems, links to documents that reference the instances are added in the user interface, next to each returned instance in the query answer (Contreras, et al., 2004), but neither the instances nor the documents are sorted by relevance. Maedche et al do provide a criterion for query result ranking in the SEAL Portal (Maedche, Staab, Stojanovic, Studer, & Sure, 2003), but the principles on which the method is based – a similarity measure between query results and the original KB without axioms, are not clearly justified, and no experimental validation is provided.

In general, this purely Boolean vision makes sense when the whole information corpus can be fully represented as a formal knowledge base. But there are limits to the extent to which knowledge can be formalized this way. First, converting the volume of unstructured information currently available worldwide into formal ontological knowledge at an affordable cost is an unsolved problem in general. This was identified decades ago as the well-known *knowledge acquisition bottleneck* (Feigenbaum, 1997) (Feigenbaum, 1984). Second, documents hold a value of their own, and are not equivalent to the sum of their pieces, no matter how well formalized and interlinked. The replacement of a document by a bag of knowledge atoms inevitably implies a loss of information value, and it is often appropriate to keep the original documents in the system. Third, wherever ontology values carry free text, Boolean semantic search systems do a full-text search within the string values. In fact, if the string values hold long pieces of free text, a form of keyword-based search takes place in practice beneath the ontology-based query model, whereby the “perfect match” assumption starts to become arguable, and search results may start to grow in size. While this may be manageable and sufficient for small knowledge bases, without a proper ranking criterion the Boolean model does not scale properly for massive document repositories where searches typically return hundreds or thousands of results.

There are nonetheless works in this context which do explicitly consider keeping, along with the domain ontologies and KBs, the original documents in the retrieval model, where the relation between ontologies and documents is established by annotation relations. In this line, KIM (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) (Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004), and TAP (Guha, McCool, & Miller, 2003) are examples of wide-ranging achievements on the construction of high-quality KBs, and the automatic annotation of documents on a large scale. Rather than the search itself, KIM focuses on the automatic population of ontologies from text corpora, along with the annotation of the latter. In one of the latest accounts of progress of this project (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004), a ranking model for retrieval is hinted at but is not been developed in detail and evaluated. In fact, KIM relies on the Lucene[33] keyword-based IR engine for this purpose (indexing, retrieval and ranking).

On its side, TAP presents a view of the search space (specifically the Web) where documents and concepts are nodes alike in a semantic network (Guha, McCool, & Miller, 2003), whereby the separation of contents and metadata is somewhat blurred. The research in TAP gave wide attention to infrastructural aspects (e.g., deployment support for KBs and distributed queries on the Web), and the presentation of results. With regards to the retrieval models themselves, the expressive power of the query language in TAP is fairly limited compared to languages such as SPARQL and the like. Specifically, the supported capabilities are limited to keyword search within the "title properties" (marked as such in the ontology) of instances, and no ranking is provided.

Another work in this line is the one by Mayfield and Finin, which combines ontology-based techniques and text-based retrieval in sequence, in a blind relevance feedback iteration (Mayfield & Finin, 2003). Inference over class hierarchies and rules is used for query expansion, and the extension of semantic annotations. Documents are annotated with RDF triples, and ontology-based queries are reduced to Boolean string search, based on matching RDF statements with wildcards, at the expense of the expressive power for queries. It is interesting nonetheless how inference is used in this work to complete missing knowledge, ultimately relying on keyword-based search wherever the knowledge coverage by ontologies and metadata falls short.

The ranking problem has been taken up again in (Stojanovic, Studer, & Stojanovic, 2003), and more recently (Rocha, Schwabe, & Aragão, 2004). Rocha et al propose the expansion of query results through arbitrary ontology relations starting from the initial query answer, where the distance to the initial results is used to compute a similarity measure for ranking (Rocha, Schwabe, & Aragão, 2004). This method has the advantage of allowing the user to express information needs with simpler, keyword-based queries but in exchange, it is not possible to define more precise (structured) query conditions taking advantage of the vocabulary and semantic relations defined by the ontology. On their side, Stojanovic et al propose a ranking scheme for ontology triples, based on the number of times an instance appears as a term in a relation type, and the derivation tree by which a sentence is inferred (Stojanovic, Studer, & Stojanovic, 2003). These two works are thus concerned with ranking formal answers to ontology-based queries, and therefore address a complementary problem to that of ranking the documents that are annotated by these answers.

[33] http://lucene.apache.org

4.4 Classification of semantic approaches

After the brief overview and broad perspective on the evolution of semantic IR techniques in the previous sections, this section studies into further detail the different approaches developed in the field, sorting them along a set of proposed (non exhaustive) classification criteria. The list of classification criteria is shown in Table 4.1 and includes:

- **Semantic knowledge representation**: research on semantic approaches in the IR field was carried through in widely explored areas such as *Latent Semantic Indexing* (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) (Letsche & Berry, 1997) and *Linguistic Conceptualization approaches* (Gonzalo, Verdejo, Chugur, & Cigarrán, 1998) (Madala, Takenobu, & Hozumi, 1998). Such proposals are commonly based on light conceptualizations, usually considering few different types of relations between concepts, and low information specificity levels. In the last few years semantic technologies have contributed with *ontology-based proposals* (Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004) (Guha, McCool, & Miller, 2003) that consider a much more detailed and densely populated conceptual space in the form of ontology-based KBs. An obvious, immediate trade-off of these approaches is that such a rich conceptual space is more difficult and expensive to obtain, but this is being one of the major targets addressed by the SW research community, which is already providing significant results and dependable grounds to build upon (Dill, et al., 2003)(Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004).

- **Scope**: the application of semantic search has been undertaken in different environments such as the *Web* (Finin, Mayfield, Fink, Joshi, & Cost, 2005) , C*ontrolled Repositories* (Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004) or even the *Desktop* (Chirita, Gavriloaie, Ghita, Nejdl, & Paiu, 2005). Among them, we should point out by its difficulty the Web. *The Web* is an open space where the information is distributed across millions of computers; where content evolves and grows extremely fast; which extends across multiple different domains; and to which millions of users with different characteristics and purposes turn to satisfy the most diverse information needs every day. Obtaining conceptualizations to cover the meanings involved in all Web content with some degree of completeness is still an open challenge. Restricting themselves to more reduced environments, many works have been undertaken and tested over *Controlled Repositories*, where the available information is enclosed in one or few domains of knowledge such as cinema, music, sports, politics, etc. Still, extracting conceptual meanings and formally representing them for specific domains of knowledge is a high cost task. To this respect, *the Desktop* environment is somewhat easier to handle –the semantic information can be easily extracted from semi-structured contents such as e-mails, folders, etc, the diversification of users is easier to cope with at this level, and the interaction with them is more explicit.

- **Goal**: semantic retrieval approaches can be characterized by whether they aim at data retrieval or information retrieval (IR). While the majority of IR approaches always return documents as response to user requests, and therefore should be classified as *information retrieval* *models*, a large amount of ontology-based approaches return ontology instances rather than documents, and therefore may be classified as *data retrieval models*. For example, as a response to the query "films where Brad Pitt plays the leading role" a data retrieval system will retrieve a list of movie instances while an IR system will retrieve a list of documents containing information about such movies. Semantic Portals (Castells, Foncillas, Lara, Rico, & Alonso, 2004) (Castells P. F., 2004) (Contreras, et al., 2004) (Maedche, Staab, Stojanovic, Studer, & Sure, 2003) and QA systems (Lopez, Pasin, & Motta, 2005), typically provide simple search functionalities that may be better characterized as semantic data retrieval rather than information retrieval. In some systems, links to documents that reference the instances are added in the user interface, next to each returned instance in the query answer (Contreras, et al., 2004), but neither the instances, nor the documents are ranked.

- **Query**: another relevant aspect that characterizes semantic search models is the way the user expresses his information needs. Four different approaches may be identified in the state of the art, characterized by a gradual increase of their level of formality and usage complexity. In the first level, queries are expressed by means of *keywords* (Guha, McCool, & Miller, 2003). For instance, a request of information about movies where Brad Pitt plays the leading role could be expressed by a set of keywords like "Brad Pitt movies". This is the most traditional way of consultation, but also the less expressive one, since the information need is represented as a set of terms without any explicit relation between them. The next level involves a *natural language* representation of the information need (Lopez, Pasin, & Motta, 2005). In this case, the previously mentioned example could be expressed as a full (interrogative) sentence, such as "in what movies Brad Pitt plays the leading role?" This kind of query provides more information than the keyword approach since a linguistic analysis can be performed to extract syntactic information, such as subject, predicate, object and other details of the sentence. The next level in formality is portrayed by *controlled natural language* systems (Bernstein & Kaufmann, 2006) (Cohen, Mamou, Kanza, & Sagiv, 2003) where the query is expressed by adding *tags* that represent properties, values or objects within the consultation. Following the previous example the query could be expressed as "s: Actor p: name v: Brad Pitt p: leading-role s: film". This kind of query is easier to process and map to the corresponding classes, properties and values of a schema or ontology underlying the search space, thus facilitating the acquisition of the semantically related information. Finally the most formal search systems are based on *ontology-query languages* such as RDQL (Seaborne, 2004), SPARQL (Prud'hommeaux & Seaborne, 2006), etc. In this approach, the previous example could expressed as "select ?f where (?a , < name>, 'Brad Pitt') , (?a, <leading-role>, ?f)" The full expressive power of this kind of query allows the system to automatically retrieve in a highly precise way the information that satisfies the information need.
- **Content retrieved**: this feature can be refined by considering the kind of information the system retrieves in response to user queries. In approaches that aim at IR, a distinction can be observed between systems that retrieve *textual information* (Gonzalo, Verdejo, Chugur, & Cigarrán, 1998) and systems that retrieve *multimedia content* (Lay & Ling, 2006) (Tsinaraki, Polydoros, Kazasis, & Christodoulakis, 2005). In data retrieval approaches, the expressive power of the provided formal language adds an additional distinction. In our state of the art

analysis we shall observe whether the systems retrieve *XML documents* (Cohen, Mamou, Kanza, & Sagiv, 2003) or proper *pieces of ontological knowledge* (Guha, McCool, & Miller, 2003) (Lopez, Pasin, & Motta, 2005).

- **Content ranking:** as we pointed out in the introduction, the definition of ranking on ontology-based search models is currently an open research problem. Most approaches do not consider ranking query results in general, other models base their ranking functionality on traditional keyword-based approaches (Guha, McCool, & Miller, 2003) and a few ones take advantage of semantic information to generate query result rankings, but generally, KB instances rather than documents are ranked (Stojanovic, 2003).

Criteria	Approaches
Semantic knowledge representation	Linguistic conceptualization Latent Semantic Analysis Ontology-based Information Retrieval
Scope	Web search Limited domain repositories Desktop search
Goal	Data retrieval Information retrieval
Query	Keyword query Natural language query Controlled natural language query Structured query based on ontology query languages
Content retrieved	Pieces of ontological knowledge XML documents Text documents Multimedia documents
Content ranking	No ranking Keyword-based ranking Semantic-based ranking

Table 4.1 Semantic search systems classification

Following the previous classification three main trends of semantic search approaches are distinguish in literature characterized by the type and use of semantic knowledge representation:

- **Latent Semantic Analysis approaches:** these models do not use human-based language understanding methodologies. On the contrary, they use statistical models, to identify groups of words that commonly appear together, and therefore describe the same reality. These approaches are the ones farther from the semantic search paradigm.
- **Linguistic Conceptualization approaches:** these approaches are the first ones to make a step towards the real semantic search, where machines attempt to understand concepts in the same way as humans do. To do so, these approaches make use of thesauri and taxono-

mies. However, compared to ontological knowledge, these conceptualizations are very light, limiting the improvements towards the semantic search paradigm.

- **Ontology-based approaches:** Ontology-based approaches are characterized by the use of highly detailed conceptualizations in the form of ontologies and KBs. They provide formal descriptions of the meanings involved in user needs and contents. Therefore, these models have better chances to achieve the so-called semantic search paradigm.

4.4.1 Latent Semantic Analysis

In the traditional keyword-based IR approaches, the potential relations between keywords are usually ignored. Thus, the importance of a keyword in a text document typically assessed by examining the occurrence of the keyword in the document and in the collection, but disregarding the occurrence of other possibly related keywords. Latent Semantic Analysis (LSA) also referred to as Latent Semantic Indexing (LSI), goes beyond this restriction by analyzing the co-occurrence of keywords documents and in the collection as a whole. LSA considers documents that have many words in common to be semantically close, and documents with few words in common to be semantically distant. Based on the co-occurrence of keywords, and the similarity of documents, words are empirically grouped into a form of "concepts", where a concept is understood as a weighted vector of semantically related words. The method aims to take advantage of implicit higher-order structure, or "semantic structure" in the association of terms with documents.

Landauer provides a thorough description of the LSA approach, and how this technique can be used to find relationships between terms and group them into concepts (Landauer & Dumais, 1997). LSA uses singular-value decomposition (SVD), a closely related technique to eigenvector decomposition and factor analysis. It takes a large matrix of term/document (or text object) association data and decomposes it into a set of, typically 50 to 150, orthogonal factors from which the original matrix can be approximated by a linear combination. More formally, a rectangular t×d (term×document matrix X) is decomposed as:

$$\underbrace{X}_{txd} = \underbrace{T_0}_{txr}\underbrace{S_0}_{rxr}\underbrace{D_0}_{rxd}$$

where T_0 and D_0 have orthonormal columns, S_0 is diagonal, and r is the rank of X. This so called singular value decomposition of X is unique modulo certain row, column and sign permutations. If only the k largest singular values of S_0 are kept along with their corresponding columns in the T_0 and D_0 matrices, and the rest deleted (yielding matrices S, T and D), the resulting matrix $\hat{X}$ is the unique matrix of rank k that is closest to X in the least squares sense. The idea is that $\widehat{X}$, containing only the first k independent linear components of X, captures the major associational structure in X, removing noisy information. This reduced model is used to approximate the term to document association data in X. Since the number of dimensions k in the reduced model is much smaller than the number t of unique terms, minor differences in terminology are ignored. In this reduced model, the closeness of text objects is thus determined by the overall pattern of term usage, so objects can be found to be similar regardless of the specific words that are used to describe them. Object descriptions are thus defined by an approximation to word meanings, thus dampening the effects of polysemy. In particular, this means that documents that do not contain the words in a user's query may still be retrieved it if the major patterns of word usage determines a relationship between the document and the query.

The work in (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) explores the use of LSA to overcome the limitations of classic IR models regarding synonym and polysemy. Initial tests find that, while the LSA method deals nicely with the synonymy problem, it offers only a partial solution to polysemy. It helps with multiple meanings because the meaning of a word can be determined not only by considering other words in the document, but by other appropriate words in the query not used by the author of a particular relevant document. The drawback is that every term is represented as just one point in the space, so that a word with several highly distinct meanings (e.g.,, "bank") is represented as a weighted average of the different meanings. If none of the real meanings is really close to the average meaning, this results in a significant distortion. This calls for some way to detect when the multiple meanings are very distant, in order to subcategorize it by several points in the space.

Criteria	Approach
Scope	Limited repositories: LSA do not scale to large document repositories
Goal	Information retrieval
Knowledge representation	Latent Semantic Analysis with extensions
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking

Table 4.2 Approach by Deerwester et al.

Dumais provides an overview of how LSA can be improved in the IR context (Dumais, 1990), exploring the techniques that have been useful in standard vector-based retrieval methods, such as differential term weightings, relevance feedback, and the selection of the number of dimensions for the reduced space. Regarding the first approach, IDF and global entropy term weighting methods improved performance by an average of 30%. The combination of a local log and a global entropy weighting (LogEntropy) yielded an improvement of 40%. Relevance feedback improved performance by an average of 67% when the first 3 relevant documents were used, and 33% when only the first relevant document is used. With respect to the number of dimensions, performance increases dramatically up to the first 100 dimensions, where it reaches a maximum and slowly degrades after that point. The tested performance ranged from roughly comparable to 30% better than standard vector-based methods, apparently depending on the associative properties of the document set and the quality of the queries.

Criteria	Approach
Scope	Limited repositories: LSA do not scale to large document repositories
Goal	Information retrieval
Knowledge representation	Latent Semantic Analysis with extensions
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking

Table 4.3 Approach by Dumais

Later work shows that, even though LSA approaches achieve a 30% better retrieval performance than lexical search techniques, the original implementation of LSA lacked the needed runtime efficiency be useful for large databases (Letsche & Berry, 1997). To overcome this problem this work proposes a new implementation of LSI named LSI++, which supports both serial and distributed searches over large datasets. To limit the amount of memory used during the process, the number of terms and documents than can be used simultaneously is restricted. The experiments have showed that the serial implementation can run up to 6 times faster in terms of search response time, while parallel implementation runs nearly 180 times faster on large documents collections.

Criteria	Approach
Scope	Web
Goal	Information retrieval
Knowledge representation	Enhanced Latent Semantic Analysis
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking

Table 4.4 Approach by Letsche

4.4.2 Linguistic conceptualization

Linguistic conceptualization approaches aim to enhance the traditional IR techniques by the use of dictionaries, such as WordNet, which provide semantic information about terms or words. WordNet is a machine-readable dictionary developed at Princeton University (Fellbaum, 1998) (Miller G. , 1995). It covers the vast majority of nouns, verbs, adjectives and adverbs from the English language. The words in WordNet are organized in sets of synonyms called synsets. Each synset represents a concept. WordNet has a large network of 129,509 words, organized in 99,643 synsets. There is a rich set of 299,711 relation links between words, between words and synsets, and between synsets.

The use of WordNet for IR has been extensively explored in previous research (Moldovan & Mihalcea, 2000) (Gonzalo, Verdejo, Chugur, & Cigarrán, 1998) (Madala, Takenobu, & Hozumi, 1998) (Madala, Takenobu, & Hozumi, 1999) (Richardson & Smeaton, 1995) (Shuang, Fang, Clement, & Weiyi, 2004) (Vorhees, 1993) (Vorhees, 1994). The aims vary from query and document disambiguation, to the enrichment of queries with related semantic terms, or the comparison of queries with documents via conceptual distance measures. Despite all these efforts, the use of WordNet to improve retrieval effectiveness has only been clearly successful in the generation of manual annotations (Gonzalo, Verdejo, Chugur, & Cigarrán, 1998) (Vorhees, 1994), or as a complement of other semantic representation resources, such as thesauri (Madala, Takenobu, & Hozumi, 1998) (Madala, Takenobu, & Hozumi, 1999).

In (Gonzalo, Verdejo, Chugur, & Cigarrán, 1998), the text retrieval task is improved by using WordNet synsets, rather than words, as the indexing space. For the experiments reported in this work, the authors have created a manually disambiguated test collection (of queries and documents) derived from SEMCOR (Miller, Leacock, Tengi, & Bunker, 1993). This is achieved in four main steps: splitting the documents in different fragments, extending the original topic tags of the Brown Corpus with a hierarchy of subtags, writing a summary for each of the fragments, and manually annotating each of the summaries with WordNet. The goal of this work is to answer two main questions: what is the potential of WordNet to abstract text retrieval from the problem of sense disambiguation? and, what is the sensitivity of retrieval performance to disambiguation errors? In the reported experiments, the retrieval performance of text documents is improved by 29% with the use of WordNet synsets instead of words to generate the indices, which clearly proves the effectiveness of WordNet. Error rates below 30% still produce better results than standard word indexing, and between 30% and 60% error rates, the results are equivalent. One of the limitations of this approach is that queries have to be also disambiguated to take advantage of the indexing approach.

Criteria	Approach
Scope	Limited domain repositories: SEMCOR (Miller, Leacock, Tengi, & Bunker, 1993)
Goal	Information retrieval, the approach retrieves text documents
Knowledge representation	Linguistic conceptualization: WordNet synsets are used as concepts to create the search indices
Query	An initial keyword query is used, but this query has to be disambiguated and translated to WordNet synsets to take advantage of this approach
Content retrieved	Text documents
Content ranking	semantic ranking: the vector space model is applied to WordNet synsets instead of words.

Table 4.5 Approach by Gonzalo et al

In (Richardson & Smeaton, 1995) a new approach to IR is proposed based on a) computing a measure of semantic distance between words and b) using this distance to compute the similarity between queries and documents. Two different similarity functions are proposed based on WordNet: the information-based approach and the conceptual distance approach. The information-based approach approximates the similarity between two words based on the hierarchy involving both terms in WordNet. On the other hand, the conceptual approach computes the similarity between two words as the sum of edge weights in the shortest path connecting their corresponding WordNet synsets. This estimator assumes that the edges between the synsets in the knowledge base are weighted. Both approaches result in a drop of effectiveness and neither of them increases the retrieval performance.

Criteria	Approach
Scope	The Web. (experiments are done with TREC collections)
Goal	Information retrieval, the approach retrieves text documents
Knowledge representation	Linguistic conceptualization: WordNet is used to compute the similarity between queries and documents
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking based on the proposed similarity measures

Table 4.6 Approach by Richardson et al

In (Vorhees, 1994) Vorhees uses WordNet as a tool for query expansion. Experiments are based on TREC collections, in which all terms in the query are expanded by a combination of synonyms, hypernyms and hyponyms. The weights of the words contained in the original query are set to 1, and a combination of 0.1, 0.3, 0.5, 1, and 2 is used in query expansion terms. The SMART IR System (Salton, 1971) is used in the evaluation. Through this method, only the performance on short queries is improved, and no significant improvement is achieved for long queries. WordNet is also used as a tool for word sense disambiguation for text retrieval (Vorhees, 1993), but a loss of retrieval performance occurs.

Criteria	Approach
Scope	The Web (experiments are done with TREC collections)
Goal	Information retrieval, text documents are retrieved
Knowledge representation	Linguistic conceptualization: WordNet is used to expand the query
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking

Table 4.7 Approach by Vorhees

The work published in (Moldovan & Mihalcea, 2000) presents a natural language interface system to an Internet search engine which provides the following improvements: a) natural language (English) questions are supported, b) queries are expanded based on search disambiguation methods and, c) a new lexical operator is used to post-process the documents retrieved, extracting only the part of the documents that is relevant to the query. This system uses WordNet for the disambiguation of keywords in the query rather than within the documents. In this approach, each keyword in the query is mapped to its corresponding semantic form. First, similarity lists are formed for each sense of one of the words, pairing the word with its different senses. Then the pairs are searched on the

Internet, ranking the different senses by the number of retrieved hits. To refine the order of senses a method called "semantic density" is used, which measures the number of words with a semantic distance below a threshold to a pre-selected set of terms, using WordNet glosses. The results obtained by this system increase the precision and the percentage of correctly answered queries, while the amount of text presented to the user is reduced.

Criteria	Approach
Scope	The Web
Goal	Information retrieval
Knowledge representation	Linguistic Conceptualization
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking

Table 4.8 Approach by Moldovan et al

In this work (Shuang, Fang, Clement, & Weiyi, 2004), Shuang et al consider that phrases are more relevant than words to compute the similarity between a query and a set of documents. Following this idea, WordNet is used to disambiguate word senses of query terms in order better compute the similarity between the query and the documents. For adjacent query words, the following information for WordNet is extracted: synonym sets, hyponym sets, and their definitions. When the sense of a query word is determined, its synonyms, its hyponyms, its compound words and the phrases contained in its definition, are considered for possible addition to the query. This system imposes and important constraint before adding a new term to the query. A new term is added only if it is highly (positively and globally) correlated with a query term/phrase. In addition, noun-phrases in queries are classified into four types: proper names of people and organizations, dictionary phrases which can be found in dictionaries such as WordNet, simple phrases which do not have any embedded phrase, and complex phrases, which are more complicated to process. The experimental results show that this approach yields an improvement between 23% and 31% over the best TREC 9, 10 and 12 collections for short queries(title only), without using Web data.

Criteria	Approach
Scope	The Web (experiments are done with TREC collections)
Goal	Information retrieval
Knowledge representation	Linguistic Conceptualization
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking

Table 4.9 Approach by Liu et al

The approach described in (Madala, Takenobu, & Hozumi, 1998) (Madala, Takenobu, & Hozumi, 1999) takes into account why, despite being used by many retrieval systems, WordNet has not been successfully used to improve the performance of those systems. Some of the mentioned problems include: a) if two terms that should be interrelated have a different part of speech in WordNet, it is not possible to find relationships between them, b) most of the relationships between two terms are not found in WordNet, and c) many terms are missing in WordNet. To overcome such limitations two approaches are proposed in this work: enriching WordNet with an automatically constructed thesaurus, and solving the problem of polysemia by expanding the queries by those terms that are more similar to the whole set of query terms. In (Madala, Takenobu, & Hozumi, 1998) two different thesauri are constructed: the co-occurrence thesaurus, based on the co-occurrence of terms in documents, and the predicated-augment thesaurus, based on the environment (nouns, verbs, adjectives, etc) where a word usually appears. In (Madala, Takenobu, & Hozumi, 1999), the effect of the Roget's thesaurus (Chapman, 1977) is also considered as additional evidence for expansion.

Criteria	Approach
Scope	Text documents
Goal	Information retrieval
Knowledge representation	Linguistic Conceptualization
Query	Keyword query
Content retrieved	Text documents
Content ranking	Traditional IR ranking

Table 4.10 Approach by Mandala et al

4.4.3 Ontology-based approaches

The Semantic Web trend has emerged with the aim of helping machines process information, enabling browsers or other software agents to automatically find, share and combine information in consistent ways. At the core of these new technologies, ontologies are envisioned as key elements to represent knowledge that can be understood, used and shared among distributed applications and machines. In this sense, ontology-based information retrieval systems are envisioned as a formal approach to semantic search.

Rocha et al (Rocha, Schwabe, & Aragão, 2004) present a search system that combines IR techniques with constrained spreading activation methods applied to a domain ontology. The system is focused on applications where the user searches for ontology instances instead of searching for arbitrary Web pages. The query language proposed in this approach is based on keywords, whereby the main goal of the system is to map those keywords to an initial set of ontology entities, and expand the results by constrained spreading activation techniques over the ontology. The process involves two main steps. The first consists of the generation of the search space, which is composed of a) a domain ontology, b) a set of weights that define the importance of the ontological relations in that domain, and c) an instance graph where each node is formed by a string that contains the instance URI and the concatenation of all the values of its properties. To compute the domain dependent property weights, three different approaches are proposed: a cluster measure, a specific measure, and a combined measure. The second step is focused on the retrieval task. The query is expressed as an initial set of keywords to be searched in the instance graph, retrieving a first set of ranked instances that fulfill the query. Given this initial set of instances and the corresponding ratings or initial activation values, the spreading activation techniques are used to find related nodes in the ontology. No ranking techniques are provided.

Criteria	Approach
Scope	Limited domain
Goal	Data retrieval
Knowledge representation	Ontologies
Query	Keyword query
Content retrieved	Pieces of ontological knowledge, mainly instances
Content ranking	No ranking is provided

Table 4.11 Approach by Rocha et al

Zhang et al (Zhang, Yu, Zhou, Lin, & Yang, 2005) propose an enhanced model that fully utilizes both textual and semantic information for searching in semantic portals. The model extends the search capabilities of existing methods and answers more complex search requests. It defines a fuzzy Description Logics IR model, and uses ontologies as background information. Given the portal KB, the portal is searched by means of formal queries. A query is modeled as a concept Q in Description Logics. Answers to the query are individuals of the concept Q which can be retrieved using the Description Logics instance retrieval algorithm. To improve this approach with IR techniques, a textual representation is assigned to each node, by considering their closest relationships in the graph. The system also accepts different kinds of queries. For keyword-based queries, the model is simplified to a traditional IR system, enhanced with the capability to search in non-document individuals of the ontology. For formal queries, only non-document instances are retrieved as query answers. Between these two extremes, the model supports different forms and degrees of integration of keyword-based queries, formal queries, and reasoning.

Criteria	Approach
Scope	Limited domain
Goal	Data retrieval and information retrieval
Knowledge representation	Ontologies
Query	Controlled natural language query
Content retrieved	Pieces of ontological knowledge and text documents
Content ranking	Traditional IR ranking

Table 4.12 Approach by Zhang et al

Cohen et al (Cohen, Mamou, Kanza, & Sagiv, 2003) propose XSEarch, a semantic search engine for XML documents, which focuses on solving the current drawbacks of search over XML documents. In the proposed approach, the user formulates queries that explicitly contain metadata, and the system returns specific XML fragments instead of entire documents as a response. A query has the form Q(S) where S = t_1, … , t_m is a sequence of required and optional search terms. A search term has the form l : k, or l : or : k; where l is a label and k is a keyword. A search term may also have a plus sign which means that it must appear in the document; otherwise, it is considered an optional term. The search space is formed by the set of XML documents. These XML documents are represented as trees where each interior node is associated with a label and each leaf node is associated with a sequence of keywords. Retrieval is achieved as follows. Let n be an interior node in a tree T. n satisfies the search term l : k if n is labeled with l and a descendent in T contains the keyword k. Once the initial set of nodes has been extracted, several techniques over the trees extend the results with meaningfully related nodes, using the interconnection relationship. To rank the retrieved fragments the weight of the different terms is computed using TF-IDF techniques. The final similarity measure between the query and the retrieved fragments is computed using the vector-space model and refined by considering the size of the fragments and the number of pairs of the fragment that contain an ancestor-descendent relationship.

Criteria	Approach
Scope	Web, XML documents
Goal	Data retrieval
Knowledge representation	XML documents
Query	Controlled natural language query
Content retrieved	Pieces of XML documents
Content ranking	TF-IDF and application of an adaptation of the vector-space IR model, enhanced using characteristics of the retrieved fragments, such as the size and the XML node hierarchy

Table 4.13 Approach by Cohen et al

Chirita et al (Chirita, Gavriloaie, Ghita, Nejdl, & Paiu, 2005) explore the application of semantics for searching in the desktop. Their research a) extracts information from user activity and information in the system, such as e-mails, folder structure, and Web cache; b) stores this context information explicitly as RDF metadata and; c) implements sophisticated semantic search functionalities on the desktop. Firstly, the system crawls and stores the semantic information extracted from the different contexts. Secondly, a full-text index is generated with all this information. Finally the search module combines keyword search on the full-text index with semantic search on the metadata repository to provide improved functionalities for finding information on the PC, to enrich the search results, and to visualize the existing contexts, using the additional knowledge stored in the metadata repository. Comparing the possibilities of a semantic desktop search to semantic search on the Web, Chirita et al conjecture that Semantic Web technologies might ultimately be more important on the desktop than on the Web. This is because the desktop environment is more limited and controlled, in the sense that most relevant contexts are described rather easily, and thus complete ontologies and metadata for the desktop environment can be easily provided. The amount of data and metadata is also more bounded in desktop environments.

Criteria	Approach
Scope	Desktop search
Goal	Information retrieval
Knowledge representation	Ontologies describing different desktop contexts
Query	Keyword query with options to restrict the search context
Content retrieved	Text documents and desktop resources
Content ranking	No ranking is provided

Table 4.14 Approach by Chirita et al

Davies et al (Davies, Weeks, & Krohn, 2002) propose a semantic search system combining traditional keyword querying with the ability to browse and query against RDF annotations. This approach tries to overcome the limitations of semantic incompleteness by providing traditional free text search when not enough metadata are available. It also combines keyword-based searches with browsing capabilities over the ontologies, allowing users to refine their queries using semantically related information. This work provides some interesting considerations such as: a) the inability of ontologies to replace the original content of the documents; b) the limitation of knowledge incompleteness in the ontologies, since all possible uses or perspectives on data can never be enumerated in advance, and c) the necessity to use metadata to increase search precision, complemented with traditional keyword-based search to increase recall. To develop the search space, RDF(S) is used to define and populate the domain ontologies. The resulting RDF annotations are then indexed along with the full text of the annotated resources. On each session, the first query is expressed using keywords, but this query is later refined by allowing the user to navigate over the ontology. The tool provides simple ranking capabilities based on traditional based search, but no evaluation is reported.

Criteria	Approach
Scope	Limited repositories
Goal	Information retrieval
Knowledge representation	Ontologies
Query	Keyword query with query refinement using ontologies
Content retrieved	Text documents containing ontological information
Content ranking	Simple ranking based on TF-IDF

Table 4.15 The QuizRDF approach

TAP is proposed as a Web-based search system where documents and concepts are nodes alike in a semantic network (Guha, McCool, & Miller, 2003). This work views the Semantic Web as a big network containing resources corresponding not just to media objects (such as Web pages, images, audio clips, etc.) as the current Web does, but also domain objects like people, places, organizations, and events. This vision is complemented with multiple relations between resources rather than just one kind (hyperlinks). These resources and their relationships are described in an RDF representation, where explicit, embedded annotations are added to link resources with the corresponding documents where they appear. The system relies on the Google Web search engine to carry out keyword-based searches. To extract the semantic information related to search results, the ontology can be queried by three different methods:

- GetData: a semi-structured query allowing Semantic Web applications to consume this semantic information. In this method, concepts and relations are expressed as: GetData(<resource>, <property>) => value.
- Search: a string is taken as input, and all the resources that contain the string in their "title property" are returned.
- Reflection: similar to the reflection methods provided by object oriented languages, returns a list of incoming and outgoing arcs of a node.

Once the query is sent against the ontology and the corresponding pieces of ontological knowledge are retrieved, the latter are augmented with data from surrounding nodes. Finally, the system presents the documents retrieved by Google and complements them with the semantically related information extracted from the ontology, which is enhanced using relationships between nodes. The semantic search capabilities are limited by the kind of queries, and no ranking is provided.

Criteria	Approach
Scope	Web
Goal	Data retrieval
Knowledge representation	Ontologies in RDF
Query	Controlled natural language query using GetData, and keyword query
Content retrieved	Pieces of ontological knowledge and text documents
Content ranking	No ranking is provided

Table 4.16 The TAP approach

The research in (Finin, Mayfield, Fink, Joshi, & Cost, 2005) (Mayfield & Finin, 2003) constitutes and important contribution to how semantic search is envisioned in the Web. It is argued in this work that for semantic Web documents or annotations to have an impact, they have to be compatible with Web-based indexing and retrieval technologies currently in place. According to this, the Semantic Web will contain two kinds of documents: a) text documents enriched by annotations in machine understandable markup, and b) documents where the content is entirely encoded in an RDF-based markup language, referencing and describing the content of conventional Web documents.

IR over collections of such documents raises new challenges and research opportunities. In these works a novel framework is proposed and analyzed over three different prototypes to explore how this Web documents enhancement by the use of metadata information can help to improve the current information retrieval process. The framework aims to explore the integration between search and inference, with the following requirements: a) support both retrieval-driven and inference-driven processing; b) retrieval should be able to use words, semantic markup, or both as indexing terms; c) Web search should rely on today's text-based retrieval engines; and d) inference and retrieval should be coupled.

Because the framework relies on traditional Web search engines it can not use the output of an inference engine as a search query. The semantic markup query should be encoded as a text query recognizable by a search engine. This process is named *swangling* for "Semantic Web mangling". To enhance the search process, text can be included in the framework. First, a text query can be sent directly to the search engine (augmented with swangled markup, if such is available). Second, the extractor can pull text, as well as markup, out of the retrieved pages.

Three prototype systems are built in this work to test the proposed approach.

- OWLIR: takes text documents, annotated with semantic markup and indices them in a custom IR system.
- Swangler: annotates ontologies with additional RDF statements, attaching terms that are indexable by standard Internet search engines.
- Swoogle: a crawler-based indexing and retrieval system for ontologies.

Swangler was designed to enable Google and other Internet search engines to index semantic Web documents. OWLIR and Swoogle, on the other hand, use special-purpose retrieval engines adapted to index and retrieve documents with RDF markup.

Criteria	Approach
Scope	Web
Goal	Data and information retrieval
Knowledge representation	Ontologies
Query	Keyword query and controlled natural language query
Content retrieved	Web documents and semantic metadata
Content ranking	Provided by traditional search engines

Table 4.17 Approach by Finin et al

Mayfield and Finin propose OWLIR (Shah, Finin, Joshi, Cost, & Mayfield, 2003), a system that retrieves documents containing both free text and semantically enriched markup. It provides a framework capable to extract and exploit the semantic information from these documents, to perform sophisticated reasoning, and to filter the results for better precision. OWLIR contains four main components: a) a set of ontologies, encoded using DAM + OIL, for information about events in a university; b) an information extraction system; c) a inference system; and d) a hybrid information retrieval mechanism. The information extraction system is used for the extraction of key phrases and elements form free text documents, which are transformed to RDF triples. These triples are used subsequently to enrich Web documents with semantic markup. They are also used by the inference system to infer additional semantic relations, by reasoning over the ontology instances and the ontology hierarchy. DAMLJessKB is used here to provide basic facts and rules that are enhanced for domain-specific proposes. The inferred semantic markup is also used to enrich the Web documents. The system has been evaluated over three different types of documents: text only, text with semantic markup, and text with semantic markup augmented by inference. The queries combined text and markup, and the results retrieved by the system include semantic information, as in a query answering system, and text documents, as in an IR system. However, no final ranking is provided.

Criteria	Approach
Scope	Web, but limited in practice by the domain ontology
Goal	Data and information retrieval
Knowledge representation	Domain ontology
Query	Controlled natural language query
Content retrieved	Semantic information and text documents
Content ranking	No ranking is provided

Table 4.18 The OWLIR approach

4.5 Discussion

This chapter revises the issue of using semantic and domain knowledge in Information Retrieval (IR), seeking for a comprehensive perspective by revisiting the work in the IR field since the early days up to the latest prospects arising from the area of semantic-based technologies. This chapter aims thus to outline a panoramic view of the area, relating concurrent yet often unconnected efforts, identifying problem variants, key distinctive dimensions in the different approaches, past and current achievements, main difficulties and open directions ahead.

We have carried out a detailed analysis of the identified drawbacks and limitations of semantic search approaches according to the proposed classification criteria. A brief summary of these limitations is presented in Table 4.19 where the last two columns identified if the limitation is suffered by the IR and the semantic-based knowledge technologies approaches respectively.

Considering *Knowledge representation* criterion, some approaches provide a **shallow representation of the information space**, equivalent in essence to the taxonomies and thesauri used before the Semantic Web was envisioned. Even though these approaches have brought improvements over classic keyword-based search; they do not exploit the full potential of an ontological language, beyond those that could be reduced to conventional classification schemes.

Considering *the scope* criterion, ontology-based approaches generally suffer from important difficulties to face large-scale and heterogeneous environments (e.g., the Web). The fundamental hurdle is the difficulty and cost of building and maintaining rich semantic resources, which commonly introduces a domain restriction. The **domain restriction** may be identified by the use of just one specific domain ontology at a time (Bernstein & Kaufmann, 2006) (Cimiano, Haase, & Heizmann, 2007) (Rocha, Schwabe, & Aragão, 2004), the use of a set of a priori defined ontologies covering one specific domain (Guha, McCool, & Miller, 2003), or the use of one large ontology which covers a limited set of domains (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004). As a result, it is impossible to scale those models to heterogeneous environments (e.g., Web), where a potential unlimited set of topics must be covered to successfully retrieve all the available information. Another important limitation of semantic search is the **scalability limitations suffered by automatic annotation methods** when applied to large-scale environments. Annotation is understood as the process of enriching data in the form of unstructured content (textual documents, images videos, etc) with semantic metadata coming from ontologies and KBs. Most of the systems require human interaction or supervision (Motta, Margas-Vera, Domingue, Lanzoni, Stutt, & Ciravegna, 2002), which constitutes one of the main bottlenecks of large-scale annotation processes. Other systems require insertion or embedding of annotations within the data (Guha, McCool, & Miller, 2003). This is clearly a non-realistic approach since Web security measures do not allow content to be modified by external programs. Finally, most of the current annotation approaches are based on a limited set of predefined ontologies and/or domains (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) which constitutes an obvious limitation for open heterogeneous environments.

Considering the *goal* criterion, ontology-based QA approaches (Lopez, Motta, & Uren, 2006) and semantic portals (Maedche, Staab, Stojanovic, Studer, & Sure, 2003) understand semantic search as a data retrieval model (a model based on an ideal view of the information space as consisting of non-ambiguous, non-redundant, formal pieces of ontological knowledge). In this view, **the information retrieval problem is reduced to a data retrieval task**. A knowledge item is either a correct or an incorrect answer to a given information request. Thus search results are assumed to be always 100% precise, and there is no notion of approximate answer to an information need. This model makes sense when the whole information corpus can be fully represented as an ontology-driven KB. However, converting the huge amount of information available worldwide, in the form of unstructured text and media documents, into formal ontological knowledge at an affordable cost is currently an unsolved problem in general.

Considering the *formalization of the query* criterion we have realized that, the systems which demand a high formalization of queries tend to be **impractical from a usability point of view**. Some systems require the user to express his needs using ontology-based query languages (Zhang, Yu, Zhou, Lin, & Yang, 2005). Other models require the user to select the ontologies beforehand (Bernstein & Kaufmann, 2006). Some platforms require the use of tedious forms (Davies, Weeks, & Krohn, 2002), and other approaches demand an excess of user feedback to interpret the query (Lopez, Pasin, & Motta, 2005). As a consequence, the effort and expertise demanded from the user makes the search process a complex and tedious task. On the other hand, it can be argued that increasing the expressivity of queries helps to improve the quality of results. A trade-off between usability and query expressivity should be achieved to encourage the use of semantic search models to non-expert users.

Considering the *retrieved content* criterion we have realized that the majority of semantic search models are based fundamentally on the retrieval of textual answers. However, along with the general growth and diversification of content in different modalities, multimedia content (audio, video, images) is becoming increasingly significant in terms of volume and value. Further research should be done in the creation of semantic search models able to manage multiple types of formats.

Considering the *content ranking* criterion, ontology-based approaches commonly provide ranking over pieces of ontological knowledge (Stojanovic, Studer, & Stojanovic, 2003), but not over unstructured information items. As we previously pointed out, these methodologies cannot be adapted to large and heterogeneous environments (e.g., the Web) where the majority of content is still unstructured. On the other hand, the semantic search systems that do provide ranking over unstructured information items (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) are generally based on traditional keyword-based ranking models and **do not exploit semantic information** to improve the ranking process.

Out of the selected criteria, another two big drawbacks of conceptual-search approaches can be easily identified:

The problem of knowledge incompleteness: the difficulties and cost of building and maintaining rich semantic resources is the other well-known fundamental hurdle, already identified at the earliest steps in the field (Croft, 1986). A fundamental issue here is to discern what expectation on the detail (depth) and coverage (breadth) would be appropriate to be realistically assumed or aimed at, and how well we may cope with the remaining incompleteness beyond that point. A potential way to satisfy the latter is by means of a graceful degradation to a classic IR system which gets by without semantics when they are insufficient.

The problem of conceptual search models evaluation: While IR systems traditionally compete against each other under formal evaluation frameworks at the annual TREC conference, to our knowledge, none of the ontology-based retrieval approaches currently reported in the literature have been validated in such rigorous ways. There are no standard evaluation benchmarks or measures for ontology-based retrieval and, even more, there is not a generalized evaluation methodology.

Criteria	Limitations	IR	Semantic
Semantic knowledge representation	Do not exploit the full potential of an ontological language, beyond those that could be reduced to conventional classification schemes	X	(partially)
Scope	Do not scale to large and heterogeneous repositories of documents		X
Goal	Are based on Boolean retrieval models where the information retrieval problem is reduced to a data retrieval task		X
Query	Limited usability		X
Content retrieved	Focused on textual content: unable to manage different formats (multimedia)	(partially)	(partially)
Content ranking	Lack of semantic ranking criteria. The ranking (if provided) relies on keyword-based approaches	X	X
Additional Limitations			
Coverage	Knowledge incompleteness	(partially)	X
Evaluation	Lack of standard evaluation frameworks:		X

Table 4.19 Limitations of semantic search approaches[34]

Based on the previous studied works we mat say that IR and semantic-based knowledge approaches are are facing the same problem from different perspectives. While semantic-based technology approaches exploit deeper levels of conceptualizations and therefore are potentially more powerful to represent knowledge, they have not take advantage of the years of experience in the IR area. This thesis attempts to bridge the gap between these two communities and proposes to explore the use of ontology-based information while the retrieval problem is formulated in a way that is proper of the IR field.

[34] The last two columns of this table identify if the limitation refers to IR or semantic-based knowledge technologies approaches. Please, note that the notation (X, partially) has been used for simplification purposes. An X does not refer to all the approaches, but to the majority of the studied systems.

Part II

An ontology-based Information Retrieval model

Summary

This part of the thesis presents the proposed semantic retrieval model. A detailed description is given of how the introduction of an enhanced conceptual level into classic IR models can help to improve performance over traditional keyword-based approaches. We discuss the potential and limitations of the approach and develop further extensions for the Web environment. Detailed evaluations of the proposed model and its extensions are reported.

Chapter 5

An ontology-based Information Retrieval model

Based on the reviewed works summarized in the preceding chapter, it is our perception that the undertakings in information search and retrieval from the semantic-based technology area can take further advantage of the technologies, background, knowledge, and accumulated experience through several decades of work in the IR field tradition. Starting from this position, we have investigated the definition of ontology-based IR models, oriented to the exploitation of domain KBs to support semantic retrieval capabilities in large document repositories, stressing on the one hand the use of ontologies in the semantic-based perspective, and on the other the consideration of unstructured content as the final search space. This chapter is organized as follows: Section 5.1 presents the motivation towards the development of novel ontology-based IR models. Section 5.2 explains the proposed ontology-based IR model in detail, including the indexing, querying, searching and ranking methods. Section 5.3 provides a brief example to illustrate the retrieval model. Sections 5.4 and 5.5 present the evaluation and discussion of the results. Finally, some chapter conclusions are given in section 5.7.

5.1 Introduction

The ideal of **supporting a higher-level conceptual (computerized) understanding of contents and queries**, to overcome the limitations of keyword-based search, have become an important trend in IR and the semantic-based knowledge technologies areas. As we have seen in the previous chapter, while there have been some important contributions in this direction in the last years, the actual fulfillment of the vision is still unclear.

Most of the approaches coming from the IR area use light conceptualizations, especially at the level of relations. Similarly, some of the semantic-based technologies approaches **make partial use of the full expressive power of an ontology-based knowledge representation**, equivalent in essence to the taxonomies and thesauri used before the SW was envisioned (Christophides, Karvounarakis, Plexousakis, & Tourtounis, 2003) (Gauch, Chaffee, & Pretschner, 2003) (Guarino, Masolo, & Vetere, 1999) (Rocha, Schwabe, & Aragão, 2004). Although these approaches have brought improvements over classic keyword-based search through e.g. query expansion based on class hierarchies, it is not clear though that these techniques alone really take advantage of the full potential of an ontological language, beyond those that could be reduced to conventional classification schemes.

On the other hand, the approaches coming from the semantic-based knowledge technology area do exploit large KBs in the order of GBs or TBs (Castells, Foncillas, Lara, Rico, & Alonso, 2004) (Cristani & Cuel, 2005) **but are typically based on Boolean retrieval models, and therefore lack an appropriate ranking scheme needed for scaling up to massive information sources.** These techniques do not consider unstructured content as the final search space. On the contrary, they are based on an ideal view of the information space as consisting of non-ambiguous, non-redundant, formal pieces of ontological knowledge. This model makes sense when the whole information corpus can be fully represented as an ontology-driven KB. However, there are limits to the extent to which knowledge can or should be formalized in this way. First, because converting unstructured text and media documents into formal ontological knowledge is a high-cost process, identified decades ago as the well-known *knowledge acquisition bottleneck* (Feigenbaum, 1997) (Feigenbaum, 1984). Second, the replacement of a document by a bag of information atoms inevitably implies a loss of information value. An third, because Boolean search systems do not generally provide clear ranking criteria, without which the search system may become useless if the search space is too big.

Aiming to take a step beyond these limitations, this chapter proposes an ontology-based retrieval model meant for the exploitation of full-fledged domain ontologies and KBs, to support semantic retrieval in document repositories. In contrast to Boolean semantic search systems, in this work perspective full documents, in addition to ontology values from a KB, are returned in response to user information needs. The search system takes advantage of both detailed instance-level knowledge available in the KB, and topic taxonomies for classification. To cope with large-scale information sources, this work proposes an adaptation of the classic vector-space model (Salton G. , 1986), suitable for an ontology-based representation, upon which a ranking algorithm is defined.

The performance of our proposed model is in direct relation with the amount and quality of information within the KB. While, if ever, ontologies and metadata become a worldwide commodity, the lack or incompleteness of available ontologies and KBs is a limitation we shall likely have to live with in the mid term. In consequence, tolerance to incomplete KBs has been set as an important requirement in our proposal. This means that the recall and precision of keyword-based search shall be retained when ontology information is not available or incomplete.

5.2 Semantic retrieval framework

Fig 5.1 shows a graphical representation of our semantic retrieval framework.

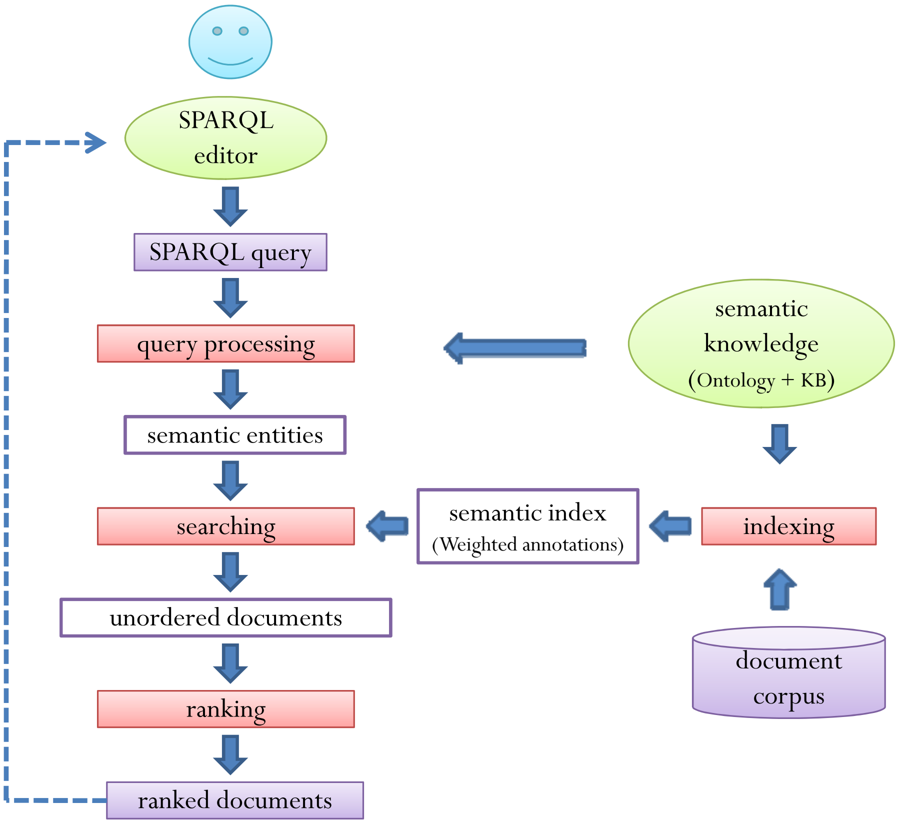


Fig 5.1 Semantic retrieval framework

As we can see in the figure, this ontology-based IR model is an adaptation of the classic keyword-based IR model described in section 2.2. It includes its four main processes: indexing, querying, searching and ranking. However, as opposed to traditional keyword-based IR models, in this approach the query is expressed in terms of an ontology-based query language (SPARQL) and the external resources used for indexing and query processing are an ontology and its corresponding KB. The indexing process is equivalent to a semantic annotation process. Instead of creating an inverted index where the keywords are associated with the documents where they appear, in the case of our ontology-based IR model, the inverted index contains semantic entities (meanings) associate to the documents where they appear. The relation or association between a semantic entity and a document is what we call annotation.

The overall retrieval process is illustrated in Fig 5.1 and consists of the following steps:

- Our system takes as input a formal SPARQL query.

- The SPARQL query is executed against a KB, which returns a list of semantic entities that satisfy it. This step of the process is purely Boolean (i.e. based on an exact match), so that the returned instances must strictly hold all the conditions in the formal query.
- The documents that are annotated (indexed) with these instances are retrieved, ranked, and presented to the user. In contrast to the previous phase, the document retrieval phase is based on an approximate match, since the relation between a document and the concepts that annotate it has an inherent degree of fuzziness.

5.2.1 Semantic indexing

In our view of semantic IR, it is assumed that a KB has been built and associated to the information sources (the document base), by using one or several domain ontologies that describe concepts appearing in the document text. This first model can work with any arbitrary domain ontology at a time with essentially no restrictions, except for some minimal requirements, which basically consist of conforming to a set of root ontology classes, which are described in the following section. The concepts and instances in the KB are linked to the documents by means of explicit, non-embedded annotations to the documents (see Fig 5.2). While we do not address in this model the problem of knowledge extraction from text (Contreras, et al., 2004) (Dill, et al., 2003) (Handschuh, Staab, & Ciravegna, 2002) (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) (Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004), we provide a vocabulary and some simple mechanisms to aid in the semi-automatic annotation of documents, once ontology instances have been created (manually or automatically).

5.2.1.1 Root ontology classes

The conceptualization of the information / knowledge space in our approach is embodied as a set of root classes. Our system requires the KB to be constructed from three main base classes: DomainConcept, Topic, and Document (see Fig 5.2).

- **DomainConcept** should be the root of all domain classes that can be used (directly or after subclassing) to create instances that describe specific entities referred to in the documents. For example, in the Arts domain, classes like Artist, Sculptor, ArtWork, Painting, and Museum should be defined as (probably indirect) subclasses of DomainConcept. A small set of upper-level open-domain classes like Person, Building, Event, Location, etc., is included in the base concept ontology, to be extended for specific domains.
- **Document** is used to create instances that act as proxies of documents from the information source to be searched upon. Two subclasses, TextDocument and MediaContent, are supplied, which can be further subclassed, if appropriate for a particular application domain, to provide for different types of documents, such as Report, News, PurchaseOrder, Invoice, Message, etc., with different fields (e.g., title, date, subject, price, sender). The class MediaContent is provided in anticipation of future extensions for multimedia retrieval. Document has a location property that holds a dereferenceable physical address (in our current implementation, a URL) from which the actual document contents can be retrieved.

- **Topic** is the root for class hierarchies that are merely used as classification schemes, and are never instantiated. These taxonomies can be any of the ones commonly associated to collections of documents, such as Open Directory Project [35](ODP) on the WWW, or the ones used in digital or physical libraries and online catalogs (e.g., the Dewey Decimal System[36]). Our system can import any such standard or application-specific classification hierarchy, by just making it available in a compatible format for our implementation (e.g., as an RDF class hierarchy). Taxonomies are used in our system as a terminology to annotate documents and concept classes, by assigning them as values for dedicated properties. For instance, in a KB for news, classes like Culture, Politics, Economy, Sports, etc. (after the IPTC Subject Reference System[37] standard), could be used as values of a (probably multivalued) topic property of the News class. Furthermore, concept classes like Athlete and Tournament could also have the topic property, in this case with the value Sports, i.e. concepts can also be classified under the same scheme as documents. Several separate taxonomies can be used simultaneously on the same documents, thus providing for multifaceted classification.

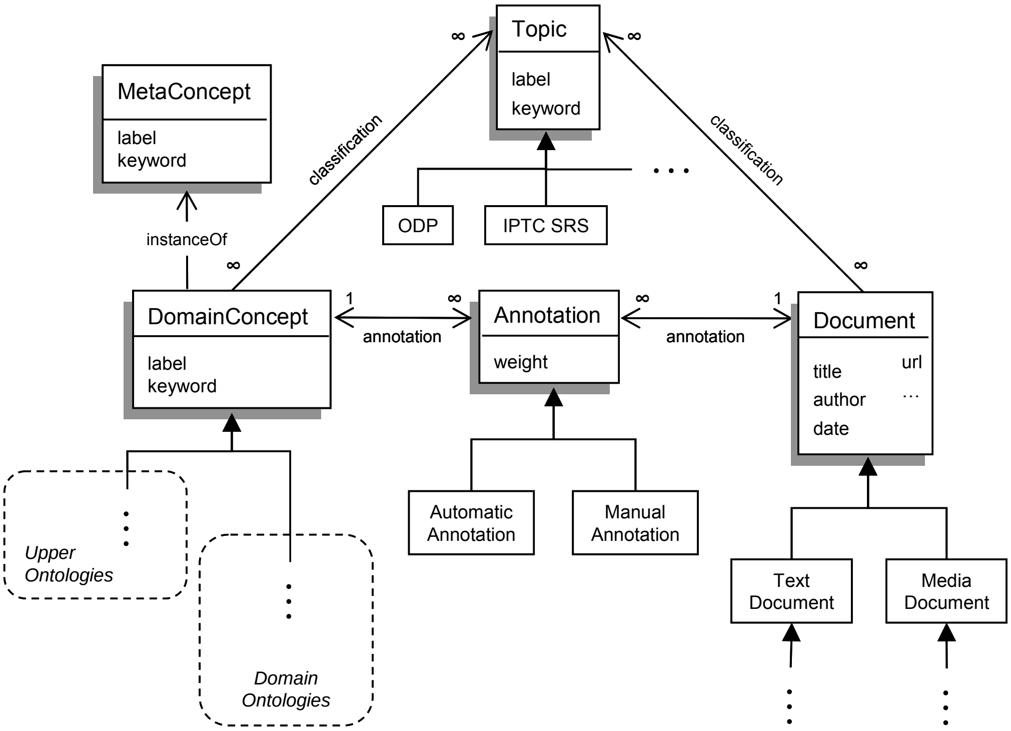


Fig 5.2 Root ontology classes.

[35] http://dmoz.org

[36] http://oclc.org/dewey

[37] http://www.iptc.org/NewsCodes

The distinction between the three root classes DomainConcept, Topic, and Document, arises from the Nets research group[38] experience in previous Semantic Web projects (Castells, Foncillas, Lara, Rico, & Alonso, 2004) (Castells P. F., 2004), This distinction allows the model to clearly separate the unstructured information (Document), the domain knowledge (DomainConcept) and the meta-information of both spaces (Topic). In our system, we exploit taxonomies for multifaceted search, and to solve word ambiguities, as will be described later.

5.2.1.2 Creating annotations

The predefined base ontology classes described above are complemented with an additional class (Annotation) that provides the basis for the semantic indexing of documents with non-embedded annotations. In many respects, this scheme for semi-automatic annotation is similar to the one reported in (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004). For convenience, annotations are represented as an extension of the ontology, but they could be implemented by any other means, as they are not a proper part of the domain knowledge. In fact, as an optimization in the implementation, the annotations are in a separate relational database. However, the availability of the annotations in ontological form has advantages such as being able to use the same tools and environments for browsing and correcting annotations as are used for editing the KB itself, and other simplifications at the implementation level. To any extent, we shall use the ontological notation here as a means to describe the structures and entities involved in the annotation of documents, and how this information is organized.

Documents are annotated with concept instances from the KB by creating instances of the Annotation class, provided for this purpose. Annotation has two main properties, instance and document, by which concepts and documents are related together. Reciprocally, DomainConcept and Document have a multivalued annotation property. Annotations can be created manually by a domain expert, or semi-automatically. The subclasses ManualAnnotation and AutomaticAnnotation are used respectively, to differentiate each case. We have found this distinction useful for the system at least because a) manual annotations are more reliable than automatic ones, and when available should prevail, and b) while automatic annotations can be deleted for recalculation, manual annotations should be preserved.

Our model provides a simple facility for semi-automatic annotation, which works as follows. *DomainConcept* instances use a *label* property to store the text form of the concept class or instance. This property is multivalued, since instances may have several textual lexical variants. Close equivalents of our *label* property are used in systems like KIM (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) (Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004) and TAP (Guha, McCool, & Miller, 2003). The value of this property can be set by hand by an ontology designer, or by semi-automatic means, if an external instance generation system is plugged to our model. An example used in our model is the automatic concept to label mapping available from the KIM KB. The semi-automatic techniques and heuristics, based on natural language processing, by which concepts are bound to

[38] http://nets.ii.uam.es

strings in KIM are described at length in (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) and (Popov, Kiryakov, Ognyanoff, Manov, & Kirilov, 2004). Similarly to KIM, once this mapping is available, instance labels are used by the automatic annotator of our system to find potential occurrences of instances in text documents. Whenever the label of an instance is found, an annotation is created between the instance and the document. In our system, documents can be annotated with classes as well, by assigning labels to concept classes.

This basic mechanism is complemented with heuristics to cope with ambiguities such as polysemy, i.e. label coincidence between different instances or classes. First the system always tries to find the longest label, e.g., “Real Madrid” is preferred to “Madrid.” The principle behind this is that a longer string is assumed to carry more specific information, which takes precedence over a more general and common meaning (indeed if a string *a* contains string *b*, then *a* occurs necessarily less or as frequently as *b*, and selecting *a* brings better accuracy in most, if not all cases). Second, classification taxonomies are used as a source of semantic context for disambiguation: a similarity measure is defined to compare the respective classification of the document and candidate synonym instances for annotation, so that the instance that has the closest classification to the document is chosen. For example, the word “Irises” in a document classified under *Arts* would be linked to an instance of *Painting* that represents Van Gogh’s famous work, rather than a subclass of *Flower*, provided that the painting instance exists in the knowledge base and has been correctly classified under *Arts*, or a taxonomic subclass thereof, and assuming that *Flower* is classified under a different taxonomic branch such as *Botany* or the like. Of course, if the *Painting* instance does not exist, our system fails because it would incorrectly annotate the document with the botanic sense. Since human supervision highly improves the accuracy of annotations, yet manually revising millions of annotations is unrealistic, after running the automatic annotation process, the system presents a reasonably short list of most uncertain annotations, to be confirmed or rejected by a domain expert. These include, for instance, unsolved polysemies, and annotations where the concepts and the documents do not have any common classification category, an indication that the right concept corresponding to the proper sense of a word might be missing from the KB.

5.2.1.3 Weighting annotations

The annotations are used by the retrieval and ranking module. The ranking algorithm is based on an adaptation of the classic vector-space model (Salton G. , 1986). In the classic vector-space model, keywords appearing in a document are assigned weights reflecting that some words are better at discriminating between documents than others. Similarly, in our system, annotations are assigned a weight that reflects how relevant the instance is considered to be for the document meaning. Weights are computed automatically by an adaptation of the TF-IDF algorithm (Salton G. , 1986), based on the frequency of occurrence of the instances in each document. More specifically, the weight d_x of an instance *x* for a document *d* is computed as:

$$d_x = \frac{freq_{x,d}}{max_y freq_{y,d}} \cdot log \frac{|D|}{n_x}$$

where $freq_{x,d}$ is the number of occurrences in d of the keywords attached to x, $max_y freq_{y,d}$ is the frequency of the most repeated instance in d, n_x is the number of documents annotated with x, and D is the set of all documents in the search space.

The number of occurrences of an instance in a document is primarily defined as the number of times the label of the instance appears in the document text, if the document is annotated with the instance, and zero otherwise. We realized in our first experiments that quite a number of occurrences were missed in practice with this approach, since pronouns, periphrasis, metonymy, etc abound in regular written speech. Finding all the references to an individual (i.e., an instance) in free text is a very complex natural language processing problem far beyond the scope of this first approach research. Nonetheless we have achieved significant improvements by extending our labeling scheme and exploiting class hierarchies as follows.

First, further instance occurrences are found by adding more labels to instances. However, the proliferation of labels tends to introduce further polysemic ambiguities that lead to incorrect annotations. To avoid this negative effect, our system provides a separate *keyword* property to be used, in addition to *label*, for instance frequency computation, but not for automatic annotation. As a general rule, *label* should be reserved to clearly instance-specific text forms, leaving more ambiguous ones as *keywords*. Since instance occurrences are only computed in the presence of an annotation, very few or no ambiguities are caused in practice.

Also, synecdoche is a frequent rhetoric figure used to avoid repetition, where an individual is referred to by its class (e.g., “the painter”), after the individual (e.g., “Picasso”) has already appeared in the text. To cope with this, the list of textual forms (labels and keywords) of an instance is automatically expanded (just for the computation of occurrences) with the textual forms of its direct and indirect classes. This introduces a slight occurrence counting imprecision when more than one instance of the same class are annotating the same document, because the same class references are counted once for each instance. For example, if “van Gogh” and “Gaugin” are cited in the same text, a reference such as “the painter” will be inaccurately counted in our current implementation as an occurrence of both painters. However, in our experiments the improvements obtained with this technique outweigh the effect of the imprecision.

5.2.2 Query processing

Our system takes as input a formal SPARQL query. This query could be generated from a keyword query, as in e.g., (Guha, McCool, & Miller, 2003) (Rocha, Schwabe, & Aragão, 2004) (Stojanovic, 2003), a natural language query (Contreras, et al., 2004), a form-based interface where the user can explicitly select ontology classes and enter property values (Castells, Foncillas, Lara, Rico, & Alonso, 2004) (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) (Maedche, Staab, Stojanovic, Studer, & Sure, 2003), or more sophisticated search interfaces (Guarino, Masolo, & Vetere, 1999). A number of research works have undertaken the construction of easy to use user interfaces for ontology query languages (Möller, Ambrus, Dragan, & Handschuh, 2008), and we do not address this problem here.

The SPARQL query is executed against the KB, which returns a list of instance tuples that satisfy the query. This step of the process is purely Boolean (i.e. based on an exact match), so that the returned instances must strictly hold all the conditions in the formal query. Rather than proposing a new approach for this operation, we reuse state-of-the-art techniques for the execution of the formal query by a standard ontology-based query engine, such as the ones packaged with popular ontology processing libraries like Jena,[39] Sesame,[40] etc. In our implementation, we are using the Jena toolkit.

Note however that it is possible to further elaborate the approach at this point towards a non-strictly Boolean model. For instance, as a supplementary enhancement, the conditions of the query could be relaxed to improve recall when not enough results are returned. Disjunctive variants of a conjunctive query could be formed in such cases, in a way that the number of conditions held determines the ranking of the result set tuples. Query variable weights could be used here as auxiliary hints to decide how strictly or flexibly each query condition should be imposed.

The SPARQL queries supported by our model can express conditions involving domain ontology instances, document properties (such as *author*, *date*, *publisher*, etc.), or classification values. E.g., "cultural articles published by the Le Monde newspaper about European movies with Canadian actors in the cast". In classic keyword-based vector-space models for information retrieval, the query keywords are assigned a weight that represents the importance of the keyword in the information need expressed by the query, or its discriminating power for discerning relevant from irrelevant documents. Analogously, in our model, the variables in the SELECT clause of the SPARQL query can be weighted to indicate the relative interest of the user for each of the variables to be explicitly mentioned in the documents. For instance, in the previous example, the user might be interested that both the movies and the Canadian actors are mentioned in the articles, or have a higher priority for either the movies or the actors. The weights can be set explicitly by the user, or be automatically derived by the system, e.g., based on concept frequency analysis, user preferences, or other strategies (Salton, 1986). In our experiments, the weights are assigned to 1 by default. For testing purposes, the user interface for the experiments provides one slider per variable with which the weights can be manually set from 0 to 1.

Our system uses inference mechanisms for query expansion based on class hierarchies (e.g., organic pigments can satisfy a query for colorants), and rules such as one by which the winner of a sports match might be inferred from the scoring. In fact, in our current implementation, it is the KB which is expanded by adding the inferred statements beforehand. As a final output of this query process, the system returns a set of tuples that satisfy the query.

5.2.3 Searching and ranking

As we described in the previous section, the query execution returns a set of tuples that satisfy the query. It is the searching module's task to obtain all the documents that correspond to the instance tuples. If the tuples are only made up of instances of domain concepts, the retriever follows all outgoing annotation links from the instances, and collects all the documents in the repository that are annotated with the instances. If the tuples contain instances of document classes (because the query

[39] http://jena.sourceforge.net

[40] http://www.openrdf.org

included direct conditions on the documents), the same procedure is followed, but restricted to the documents in the result set, instead of the whole repository.

Once the list of documents is formed, the search engine computes a semantic similarity value between the query and each document, as follows. Let O be the set of all classes and instances in the ontology, and Δ be the set of all documents in the search space. Let $q \in \Theta$ be an SPARQL query, let V_q be the set of variables in the SELECT clause of q, let w be the weight vector for these variables, where for each $v \in V_q$, $w_v \in [0,1]$. Let $T_q \subset O^{|V_q|}$ be the list of tuples in the query result set, where for each tuple $t \in T_q$ and each $v \in V_q$, $t_v \in O$.

We represent each document in the search space as a document vector $d \in \Delta$, where d_x is the weight of the annotation of the document with concept x for each $x \in O$, if such annotation exists, and zero otherwise. We define the extended query vector[41] q as given by $q_x = \sum_{\exists t \in T_q, t_v = x} w_v$, i.e. the query vector element corresponding to x is added the variable weight w_v if there is a tuple t where $t_v = x$ (even if there is more than one such tuple, w_v is not added more than once for the same v and x). Note that the sum rarely has more than one term, since this would mean that the same instance appears as a satisfying value for different variables in different (or the same) result set tuples. If x does not appear in any tuple, we set $q_x = 0$. Now the similarity measure between a document d and the query q is computed as:

$$sim(d,q) = \frac{d \times q}{|d| \cdot |q|}$$

Because of the way q is constructed, $|q|$ is usually quite large, and the values of $sim(d,q)$ are quite low. For example, if the user queries for special offers for summer holidays in the Aegean Islands, it can be seen that a document that shows one such offer will get a similarity value in the order of $1/n$, where n is the total number of registered offers in the knowledge base that match the query. Only a document that would display nearly all offers could get close to similarity 1. This potential problem is solved by a normalization of the similarity scores that is part of the following step.

5.2.4 Dealing with the problem of knowledge incompleteness

If the knowledge in the KB is incomplete (e.g., there are documents about travel offers in the knowledge source, but the corresponding instances are missing in the KB), the semantic ranking algorithm performs very poorly: SPARQL queries will return less results than expected, and the relevant documents will not be retrieved, or will get a much lower similarity value than they should. As limited as might be, keyword-based search will likely perform better in these cases. To cope with this, our ranking model combines the semantic similarity measure with the similarity measure of a keyword-based algorithm.

[41] Without loss of generality, we shall use the same symbol for the query and the corresponding query vector. Likewise, we identify a document with its document vector

Combining the output of several search engines has been a widely addressed research topic in the IR field (Croft, 2000) (Lee J. H., 97). After testing several approaches we have selected the so-called CombSUM strategy, which has also been found to be among the most simple and effectives in prior work, and consists of computing the combined ranking score by a linear combination of the input scores. That is, in our case the final score is $\lambda\, sim\,(d,q) + (1-\lambda)\, ksim\,(d,q)$, where $ksim$ is computed by a keyword-based algorithm, and $\lambda \in [0,1]$. We have taken $\lambda = 0.5$, which seems to perform well in our experiments. As a further adjustment, if $ksim$ returns 0, we take $\lambda = 1$, and if sim returns 0, we take $\lambda = 0.2$. For further testing, we have implemented a user interface where this parameter can be freely set by the user with a slider after the search has been executed, so that the user can see dynamically how the results are re-ranked as the value of λ is moved. Obviously, for the combination of scores to make sense, the scores have to be first made comparable, which involves a normalization step. For this purpose, we use our own optimized normalization method (explained in chapter 8), which not only scales the scores to the same range (the [0,1] interval) as other standard approaches proposed in the literature do (Lee J. H., 97), but moreover undoes potential biases in the distribution of the scores.

The automatic creation of a keyword-based query, to be combined with the results of the semantic query, remains to be explained. The keywords for the $ksim$ algorithm could be extracted directly from the user query, if a keyword-based or even natural language interface is used. In our current implementation, we extract the keywords from the SPARQL query, which is suitable enough for our present tests, and would be appropriate for a form-based query interface as well. More specifically, the value of the *label* property of a) the class of all query variables for which a *rdf:type* clause is included in the query, and b) any instances explicitly appearing within the SPARQL query, are taken as query keywords. For example, the following query would yield the query keywords "company," "Food, Beverage & Tobacco," "located in," "USA," "net income," "greater than," "3000000."

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX kb:   <http://nets.ii.uam.es/>
SELECT        ?company
    WHERE    { ?company  rdf:type kb:Company.
               ?company kb:activeInSector kb:FoodSector.
               ?company kb:locatedIn kb:USA.
               ?company kb:income ?income .
                          FILTER (?income > 3000000). }
```

In sum, our method improves keyword-based search (actually outperforms it, as is shown in the section 5.4) when the relevant information is available in the KB, and relies on keyword-based search otherwise.

5.3 Example

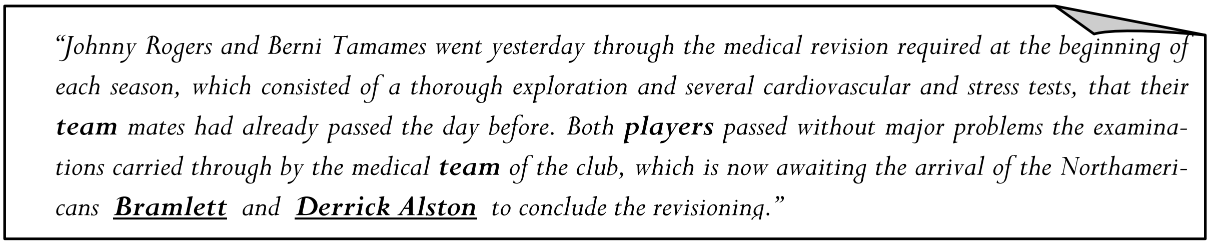
In order to illustrate our model, consider the query example: "Players from USA playing in basketball teams of Catalonia." This would be formalized as:

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX kb:  <http://nets.ii.uam.es/>
              SELECT        ?player ?team
                 WHERE        { ?player rdf:type kb:SportsPlayer.
                                 ?player kb:plays kb:Basketball.
                                ?player kb:nationality kb:USA.
                                ?player kb:playsIn ?team.
                                ?team kb:locatedIn kb:Catalonia.}
```

Assume that in order to give higher priority to the players themselves, a weight of 1.0 is assigned to the variable *?player*, and a weight of 0.5 to the *?team* variable. We have run this query against a reduced sample document set taken from a regional Spanish newspaper archive, using a small KB containing knowledge about sports in Spain (see section 5.4 for the details of how this KB was set up), which returns the following tuples:

Player	***Team***
Aaron Jordan Bramlett	*Caprabo Lleida*
Derrick Alston	*Caprabo Lleida*
Venson Hamilton	*DKV Joventut*
Jamie Arnold	*DKV Joventut*

Therefore, the query vector *q* has a value of 1 for the vector-space "axis" defined by the four player instances, and a value of 0.5 for the two teams.



"Johnny Rogers and Berni Tamames went yesterday through the medical revision required at the beginning of each season, which consisted of a thorough exploration and several cardiovascular and stress tests, that their ***team*** *mates had already passed the day before. Both* ***players*** *passed without major problems the examinations carried through by the medical* ***team*** *of the club, which is now awaiting the arrival of the Northamericans* ***Bramlett*** *and* ***Derrick Alston*** *to conclude the revisioning."*

Fig 5.3 First result for "Players from USA playing in basketball teams of Catalonia."

The second step of the retrieval algorithm retrieves 66 news articles ranked from 0.1 to 0.52. As an example, the document in the top of the result list is shown in Fig 5.3. The document is annotated by the instances that represent the players *Aaron Bramlett* and *Derrick Alston*, shown in underlined italic in the text. The weights computed for the annotations are 1.73 and 1.65, respectively, and thereby, the document vector *d* has these values in the coordinates that correspond to the players, and 0 anywhere else. The resulting rank value for the document is sim(d,q) = 0.12, and the score computed by the keyword-based algorithm is ksim(d,q) = 0.06, resulting from the occurrence of the keywords shown in bold in the text. These scores are normalized to 0.63 and 0.41 respectively, so that the combined rank value is 0.52.

5.4 Evaluation

In contrast to traditional IR communities, where evaluation using standardized techniques, such as those described in section 2.4, has been common for decades; the SW community is still a long way from defining standard evaluation benchmarks that comprise all the required information to judge the quality of ontology-based retrieval approaches. Current approaches for SW technologies evaluation are based on user-centered methods. (Sure & Iosif, 2002) (McCool, Cowell, & Thurman, 2005) (Todorov & Schandl, 2008). These evaluation techniques involve users to judge the quality of SW applications under specific use cases. Therefore, they tend to be high-cost, non-scalable and difficult to repeat. Attempting to give a step towards these limitations we have decided to generate a medium-scale IR-based evaluation benchmark for semantic retrieval approaches.

5.4.1 Evaluation benchmark

The evaluation benchmark comprises a text collection, a set of queries and their corresponding document judgments, ontologies that cover the query topics and KBs that populate the ontologies, preferably using a source independent of the text collection.

- **The Document Collection**: We decided to construct a benchmark taking a corpus of 145,316 documents (445 MB) from the CNN Web site. We chose the CNN Web site because it contains news about a wide variety of topics. The document corpus was scrapped from the Web using a wrapper-based approach as the one described in section 3.3.2. For each document we stored the title, the subtitle, the date and the related topics.
- **The Ontologies and Knowledge Bases**: We have used the KIM domain ontology and KB (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004), publicly available as part of the KIM Platform, developed by Ontotext Lab[42], with minor extensions and adjustments to conform to our top-level meta-model described in Section 5.2.1.1. We have also manually added classes and instances in areas where the KIM KB fell short (such as the Sports domain), in order to support a larger testbed for experimentation. Only one classification taxonomy is used, based on the categories of the CNN archive (such as Business, Politics, Sports, etc., and subcategories thereof), with which all documents and domain classes are classified, as explained in Section 5.2.1.1. Our implementation is compatible with both RDF and OWL. The complete KB includes 281 classes, 138 properties, 35,689 instances, and 465,848 sentences, taking a total of 71 MB in RDF text format. For efficiency, the KB has been stored on a MySQL back-end using Jena 2.2. Based on the concept-keyword mapping available in the KIM KB, over $3·10^6$ annotations (i.e. over 25 per document on average) are automatically generated by the techniques described in section 5.2.1.2.

[42] http://www.ontotext.com/kim

- **Queries and judgements** Since the semantic queries to our system can be arbitrarily complex, and should be based on the available ontologies, it is not clear how a meaningful random query generation procedure could be put in place. Therefore, a set of twenty queries was prepared manually for the comparative performance measurement. We report and discuss in the next section the observed results.

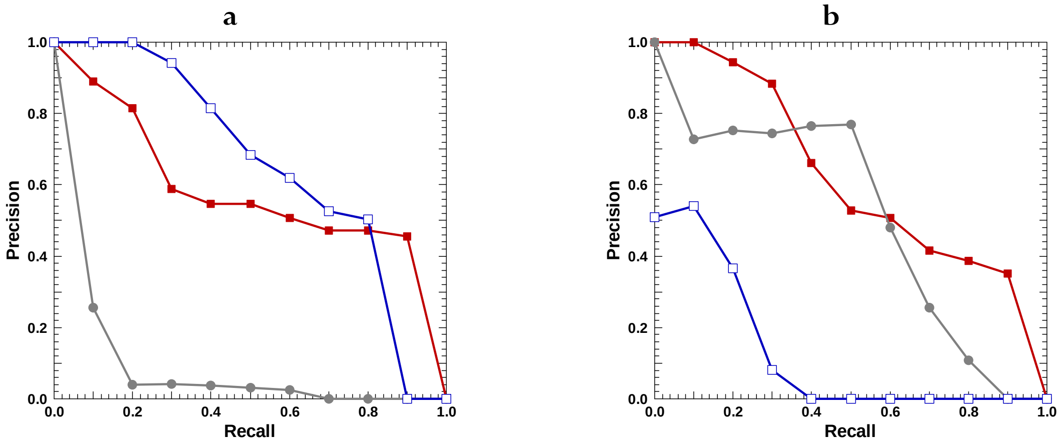
5.4.2 Experimental conditions

The experiments were designed to compare the results obtained by three different search approaches:

- **Keyword search**: a conventional keyword-based retrieval model, using the Jakarta Lucene library[43].
- **Ontology-only search**: the ontology-based retrieval model explained in this chapter without including the final step of ranking combinations.
- **Semantic search**: the completed semantic retrieval model explained in this chapter, including the combination of keyword-based and ontology-based retrieval results.

5.4.3 Results

We report and discuss in this section the observed results on three examples selected among the twenty generated queries. We show different levels of performance for different characteristic cases, where we have intentionally chosen examples where the ontology-based method does not always return the best results. The overall performance over the twenty queries is also shown on average over the whole test set. The metrics are based on a manual ranking of all documents for each query, on a scale from 0 to 5. In the experiments, all the query variables were given a weight of 1. The measurements are subjective and limited, yet indicative of the degree of improvement that can be expected, and in what cases, with respect to a keyword-based engine. The results are shown in Fig 5.4.



[43] http://lucene.apache.org

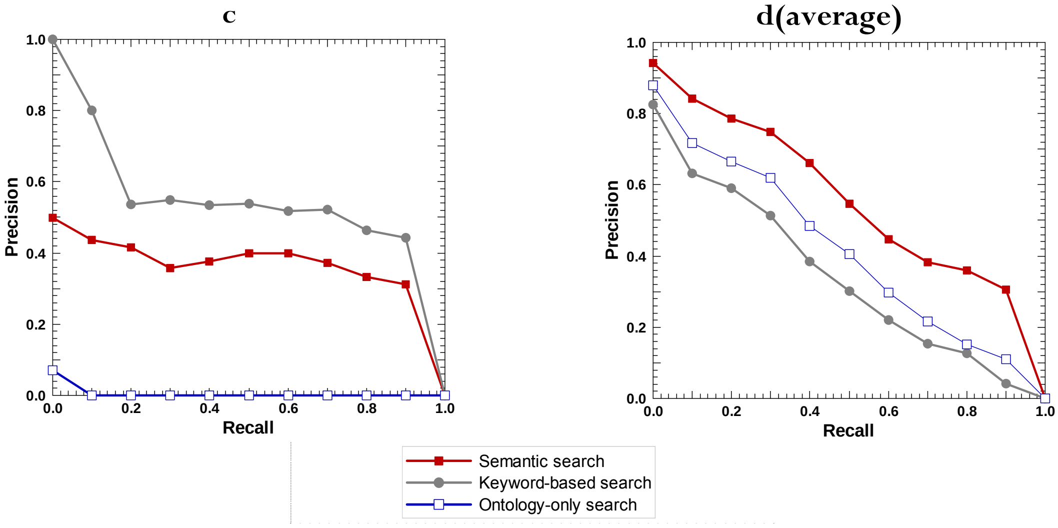


Fig 5.4 Evaluation of ontology-based retrieval against keyword-based only[44].

- **Query a.** *"News about banks that trade on NASDAK, with fiscal net income greater than two billion dollars."* In this example the semantic retrieval algorithm outperforms keyword-based search because the limited expressive power of the latter fails to express all the conditions in the query. Furthermore, the KB contains many instances of banks, some of which match the query, and news about these banks are recognized as relevant by the semantic retrieval algorithm as soon as their name is mentioned in the document, even if the text does not mention trade markets or fiscal incomes. With keyword-based search, only the documents that explicitly contain words like "bank" and "NASDAK" are ranked highly. These are typical results when a search query involves a region of the ontology with a high degree of completeness in terms of instances and annotations. These cases yield a high precision up to almost maximum recall. However, the KB does not contain all banks, which explains the decrease of precision at 100% recall. If more instances were added, precision would stand at high levels for all the recall values.

- **Query b.** *"News about telecom companies."* In this example, the ontology KB has only a few instances of telecom companies, so not all documents relevant to the query are annotated. This causes low precision values for the ontology-based approach, which drop to 0 for higher recall. The example shows how the combination of semantic and keyword-based results retains the efficiency of the latter when the former fails. Furthermore, in the areas where semantic retrieval does work (here, at low recall), the combined approach takes advantage of these few good results to perform better than the keyword-based techniques.

- **Query c.** *"News about insurance companies in USA."* This example shows a case where our method fails. The performance of the semantic retrieval is spoiled by incorrect annotations, namely, the "Kaye" insurance company is confused with "Kaye" as a person's name. Similar-

[44] The performance of the algorithms, in terms of precision vs. recall figures is shown for three different queries a, b, and c, and averaged over 20 queries

ly, the “Farmers” insurance group is incorrectly assigned as an annotation to documents where the word “Farmers” refers to farm people but has no relation with insurance. It is clear that these false positives could be considerably reduced by better information extraction techniques, beyond the scope of this paper, which would discard all different meanings of these words in a text but one, once the word “insurance” is found. In exchange, this would cause misses when e.g., the word “Kaye” appears several times in the same document with legitimately different meanings (the company vs. a person), but this case is likely to be quite rare. The problem with the word “Farmers” is also related to the fact that the concept of “Farmer” as a farm person is missing in the ontology, so it cannot even be considered as an alternative for annotation. If it was included, at least an ambiguity would be detected, and there would be a chance to solve it – even as a last resort the user might be warned. Despite these problems, and aside all the possible improvements to overcome them, it can be seen that the combination with keyword-based relevance reduces the loss of precision considerably already.

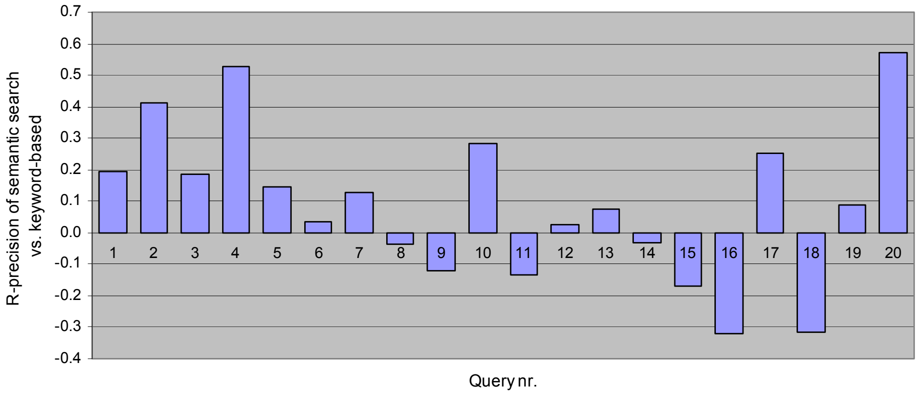


Fig 5.5 Comparative precision histogram for semantic retrieval vs. keyword-based search.

The examples described in this section are representative of the typical behavior of our techniques in characteristic cases. Situations like the one illustrated by query c, where conventional search would work better, and others where the lack of knowledge in the KB results in a loss of precision, are compensated on average by the cases where the KB has a good coverage and the annotations are accurate. Fig 5.5 shows an average comparison of the performance of our system over the set of twenty queries (which comprise the three ones a, b, and c analyzed above), and Fig 5.5 shows the difference in performance (measured by R-precision) between our approach and conventional search for each of the twenty test queries. Queries a, b, and c correspond to 2, 6, and 15 in the figure, respectively. The worst performing results in queries 16 and 18 are due, again, to incorrect annotations. This suggests that further work on the automatic annotation techniques are worthy areas for enhancing the behaviour of our model. Overall, a significant improvement achieved by our approach can be observed in the global comparison provided by the histogram and the average precision curve.

Although a systematic efficiency testing has not yet been conducted, the average informally observed response time on a standard professional desktop computer is below 30 sec. A main bottleneck in our first implementation was the traversal of annotations to retrieve the document vectors, the cost of which grows linearly with the size of the result sets ($|T_q|$ and $|R_q|$, where $R_q = \{d \in \Delta \mid \text{sim}(d, q) > 0\}$). This was drastically reduced by storing the annotations in a separate database.

5.5 Discussion

The added value of semantic information retrieval with respect to traditional keyword-based retrieval, as envisioned in our approach, relies on the additional explicit information: type, structure, relations, classification, and rules, about the concepts referenced in the documents, represented in an ontology-based KB, as opposed to classic flat keyword-based indices. Semantic retrieval introduces an additional step with respect to classic information retrieval models: instead of a simple keyword index lookup, the semantic retrieval system processes a semantic query against the KB, which returns a set of instances. This can be seen as a form of query expansion, where the set of instances represent a new set of query terms, leading to higher recall values. Further implicit query expansion is achieved by inference rules, and exploiting class hierarchies. The rich concept descriptions in the KB provide useful information for disambiguating the meaning of documents. In summary, our proposal achieves the following improvements with respect to keyword-based search:

- Better recall when querying for class instances. For example, querying for “British companies quoted on NYSE” would return documents that mention e.g., *Barclays PLC*, *Vodafone* and other such companies, even if the words “British” and “NYSE” are not present in the documents.
- Better precision by using structured semantic queries. Structured queries allow expressing more precise information needs, leading to more accurate answers. For instance, in a keyword-based system, it is not clearly possible to distinguish a query for USA players in European basket teams vs. European players in USA teams, which is possible with a semantic query.
- Better precision by using query weights. Variables with low weights are only used to impose conditions on the variables which really matter. For example, the user can search for news about USA players in European teams, regardless of whether the news mention the team at all.
- Better recall by using class hierarchies and rules. For example, a query for *WaterSports* in Spain would return results in *ScubaDiving*, *Windsurf*, and other subclasses, in *Cádiz*, *Málaga*, *Almería*, and other Spanish locations (by the transitivity of *locatedIn*).
- Better precision by reducing polysemic ambiguities using instance labels and classifications of concepts and documents.
- Despite the separation of the content space (documents) and the concept space, it is possible to combine conditions on concepts and conditions on contents. For example, in a query like “film reviews published within the current year about Japanese sci-fi movies,” the type (film review) and date (current year) requirements are set on the document, whereas the rest of the query defines conditions on some concept (a movie), not in the document space, that annotates the document.

- The improvements of our method with respect to keyword-based search increase with the number of clauses in (i.e. the specificity of) the formal query. This is not surprising, since the higher the complexity of the information need, the more query information is lost in a keyword-based query.
- As explained and shown along this paper, the degree of improvement of our semantic retrieval model depends on the completeness and quality of the ontology, the KB, and the concept labels. For the sake of robustness, the system resorts to keyword-based search when the KB returns poor results.

The combination of keyword-based and ontology-based rankings is tricky. While the inclusion of keyword-based results ensures the robustness of our method when ontology-based results are bad, this is at the expense of a precision loss in the opposite case. The employed score combination strategy, explained in chapter 8 achieves an effectiveness above the average of both techniques, in fact closer to that of the best performing model for each query, as was shown in the examples, but further investigation is worth in this area performed before the clustering processes.

5.6 Conclusions

The aim of this semantic retrieval model is to provide better search capabilities which yield a qualitative improvement over keyword-based full-text search, by introducing and exploiting finer-grained domain ontologies. Our approach can be seen as an evolution of the classic vector-space model, where keyword-based indices are replaced by an ontology-based KB, and a semi-automatic document annotation and weighting procedure is the equivalent of the keyword extraction and indexing process. We show that it is possible to develop a consistent ranking algorithm on this basis, yielding measurable improvements with respect to keyword-based search, subject to the quality and critical mass of metadata.

There is ample room for further improvement and research beyond our current results. For instance, our proposal inherits all the well-known problems of building and sharing well-defined ontologies, populating KBs, mapping keywords to concepts and annotating with documents. It is our aim to provide a consistent model by which any advancement on these problems is played to the benefit of semantic retrieval improvements. Along this line, the thesis undertakes further steps towards an effective deployment of the semantic IR approach on a decentralized, heterogeneous, dynamic and massive repository of content such as the Web.

As discussed in section 4.5 this objective implies to address several challenges such as:

- **Heterogeneity**: Our largest-scale experiments at this point are based on the KIM KB, one of the largest-sized, publicly available ontology at the time of writing. This ontology provides a reasonably good coverage of knowledge areas of general importance (geographical locations, organizations). However, the contents available on the Web describe a potential unlimited number of domains. Therefore, better levels of knowledge coverage should be reached. To address this problem we propose: a) the generation of a SW gateway that stores and provides fast access to the increasing amount of online available semantic metadata (chapter

7) and, b) the adaptation of the semantic retrieval model to exploit these large amounts of semantic information (sections 6.2.1 and 6.2.2)

- **Scalability**: Addressing scalability limitations is still an open problem of ontology-based IR approaches. The most ambitious perspectives often raise controversy with respect to their feasibility, as they indeed posit hypothesis which are difficult to grant a priori. Scaling our model to the Web environment implies, on the one hand, to exploit all the increasing available semantic metadata in order to provide a good cover of topics and, on the other hand, to manage huge amounts of information in the form of unstructured contents. To address this problem we propose the creation of scalable and flexible annotation processes that associate the Web contents with semantic metadata; maintaining the two information sources decoupled (section 6.2.1).
- **Usability**: Another important requirement in order to extend our ontology-based retrieval model to the Web environment is to provide users with an easy to use query UI (section 6.2.2). This means not to require users to have previous knowledge about ontology-based query languages, or to navigate across complex forms to formulate their queries.

In the next chapter we address the above challenges, extending and exposing the proposed model to a retrieval space with the characteristics of the World Wide Web.

Chapter 6

Semantic retrieval on the Web

The semantic search model detailed in chapter 5, as well as other semantic technologies that have proved to work well in specific domains, still have to face several open challenges in order to scale up to a massive and open search space such as the World Wide Web. The research reported in this chapter takes a step in this direction by extending the basic semantic retrieval model proposed in chapter 5 to a more scalable and flexible model, open to a huge, dynamic and heterogeneous document repository such as the Web. The chapter is organized as follows: section 6.1 motivates the problem and introduces the challenges that we should address. Section 6.2 explains the adaptations performed to our original ontology-based IR model. Section 6.3 introduces the new Web-scale evaluation benchmark and presents the experiments and results. Finally, section 6.4 provides a brief summary and a discussion of the material presented in this chapter.

6.1 Motivation

Beyond the improvements by semantic retrieval over traditional keyword-based search technologies in controlled environments (as shown in chapter 5), further study of the field indicates that while ontology-based search systems have been shown to perform well in for organizational semantic intranets (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004) (Maedche, Staab, Stojanovic, Studer, & Sure, 2003) there have not been convincing attempts at applying ontology-based search to the Web as a whole. The advancements to date are limited and partial, and can certainly not be compared to those achieved in the IR field, neither in scalability, nor in generality.

The difference between traditional IR systems and current ontology-based approaches starts in fact at the level of problem formulation. Most current ontology-based search approaches ignore the IR process as a whole, where the user expresses his information need using a set of keywords, and obtains as an answer a ranked set of documents. This difference is translated into **usability limitations** (systems demand users to have previous knowledge about ontology-based query languages or to navigate across complicate forms to formulate their queries) **and scalability limitations** (systems have an ideal vision of the information space consisting on a translation of the whole unstructured Web information corpus into formal pieces of ontological knowledge).

The other big limitation of ontology-based search approaches is their difficulty to deal with the **heterogeneity** of Web environment. The contents available on the Web describe a potential unlimited number of domains. However, semantic retrieval systems are generally limited to a predefined set of ontologies and their level of knowledge coverage is very limited.

The semantic retrieval model proposed in chapter 5 is therefore extended to address the above mentioned limitations, towards its application in the Web environment. The key features of the modified approach are:

- **Usability**: It does not require users to learn any special-purpose query language. The system supports queries expressed in natural language.
- **Scalability**: It does not require translating the whole unstructured Web information corpus into formal pieces of ontological knowledge. On the other hand, it uses both, already available relevant semantic data drawn from the SW and the information found in standard Web pages, to answer user queries.
- **Heterogeneity**: It exploits the increasing amount of semantic metadata available online, thus covering a wider and not pre-defined range of domains.

To properly evaluate the extended semantic retrieval model we need a Web-scale evaluation benchmark. While the IR community has their own standard Web-based evaluation benchmarks (see section 2.4.2.3), current ontology-based retrieval technologies still lack of formal evaluation frameworks. The work of this chapter aims to take a step beyond this problem and to propose a new potentially widely applicable benchmark for evaluating Web-oriented ontology-based retrieval systems. This benchmark is the result of the formalization of evaluation methodologies and datasets for ontology-based retrieval, drawing from the IR tradition and standard resources. To our knowledge, none of the ontology-based search approaches reported in the literature at the time of writing have been validated in such rigorous ways.

6.2 Semantic retrieval framework extensions

Fig 6.1 shows the extensions made to our original semantic retrieval framework (chapter 5).

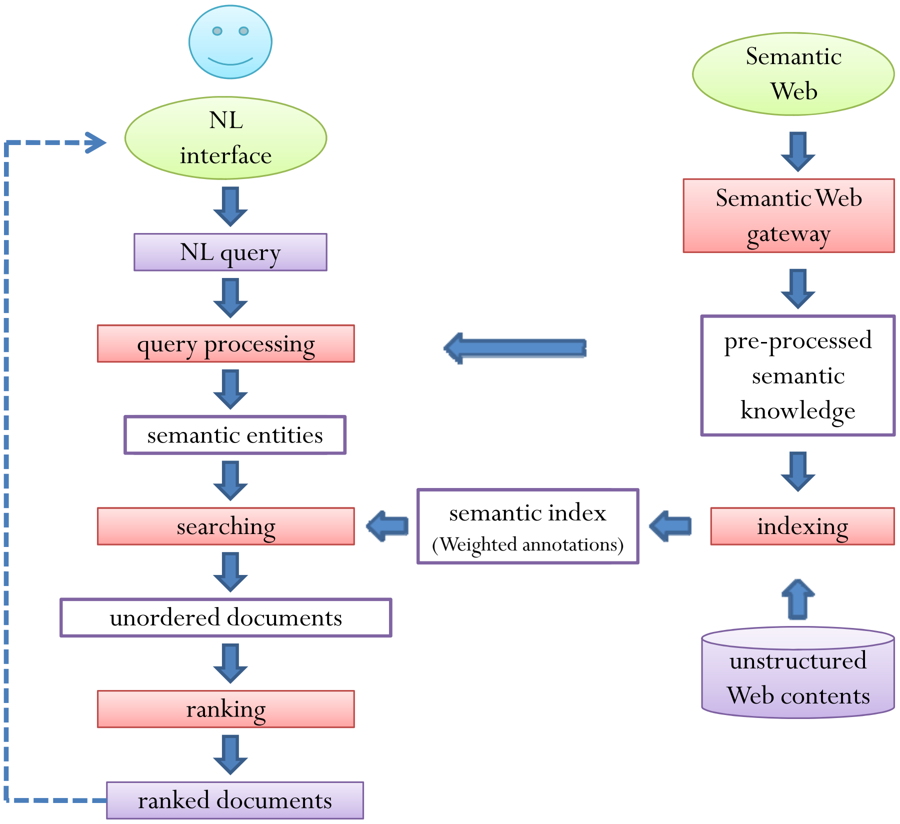


Fig 6.1 Semantic retrieval framework extensions

Three main changes can be perceived in the architecture:

- The queries are not expressed using ontology-based query languages. Instead, queries are expressed in natural language as a compromise between expressivity and usability.
- The external resources for indexing and query processing are not a single ontology and KB but online available SW information.
- In order to manage large amounts of semantic information during the query and annotation processes, a SW gateway is generated with the aim of gathering, storing and accessing the online distributed SW information (this new module is explained in detail in chapter 7).

The rest of the framework, specially the four main tasks of IR systems: document indexing (annotation), query processing, searching and ranking have also been adapted to exploit the information spaces defined by the SW and by the (non-semantic) WWW. The details of how these modules have been extended are explained in the following sections. The overall retrieval process is illustrated in Fig 6.1, and consists of the following steps:

- Our system takes as input a user's natural language (NL) query. This query is processed by the query processing module, which has been replaced by an ontology-based QA system, PowerAqua (Lopez, Motta, & Uren, 2006). This component operates in a multi-ontology scenario where it translates the user terminology into the ontologies terminology. To ensure fast access to the (online) available ontologies, it makes use the ontology indexing structures defined in section 7.2. The integration of this QA system in our model brings two clear benefits to adapt our model to the Web environment. First, the user interaction is eased by allowing natural language queries, increasing the **usability** of our system. Second, the response is obtained from a large set of ontologies covering a potential unrestricted set of domains, therefore dealing with the **heterogeneity** limitations. For example, given the query "which are the members of the rock group Nirvana?" and two ontologies covering the term Nirvana (one about spiritual stages and one about musicians), PowerAqua is able to: 1) select these two ontologies containing the term Nirvana; 2) choose the appropriate ontology after disambiguating the query using its context and the available semantic information and; 3) extract from this ontology an answer in the form of ontological entities. In this case it returns a set of individuals corresponding to the members of the group, i.e., Kurt Cobain, Dale Crover, etc. Note that the results obtained by PowerAqua are the replacement in our previous model to the answers retrieved by the SPARQL query. However, as opposed to the SPARQL query the results obtained by PowerAqua are extracted from several ontologies and KBs at a time.

- Once the pieces of relevant ontological knowledge have been returned as an answer to the user's query, the system performs a second step to retrieve and rank the documents containing this information. To do so, the document collection is automatically indexed in terms of the ontology concepts prior to the use of the system. The indexing module has been changed to integrate **scalable** and flexible annotation algorithms. This new indexing algorithms are able to deal with large document collections and large amounts of ontologies and KBs. Exploiting large amounts of metadata brings the advantage of retrieving Web documents without any potential domain restriction, therefore addressing the **heterogeneity** limitation. Continuing our previous example, in addition to the answers retrieved by the query processing module, our system performs a search for relevant documents and ranks them accordingly. The ranking algorithm, based on an adaptation of IR the vector space model, was initially designed to scale up to large document repositories, and it reminds from our previous model.

The final output of the system consists of a set of ontology elements that answer the user's question and a complementary list of semantically ranked relevant documents.

All the previous mentioned steps are carried out using five main architectural components: (1) the SW gateway, which pre-processes (online) available semantic information; (2) the query processing module, or PowerAqua module, which answers a natural language query in the form of pieces of ontological knowledge; (3) the semantic indexing module, which generates a concept-based index to link the semantic information with the Web documents; and (4) the document retrieval and ranking module, which makes use the previously generated concept-based index to retrieve and rank the Web documents relevant to the pieces of ontological knowledge previously obtained by PowerAqua.

6.2.1 Semantic indexing

At present, two different types of information coexist on the Web. On the one hand, the incipient but growing body of metadata being produced under the influence of the SW view and technologies, which delivers the capability to model explicit conceptualizations. On the other hand we have the huge amount of unstructured documents which make up the current Web content. While the semantic data has the potential of improving search, most information available on the Web nowadays is still in the form of unstructured content.

In our view of semantic retrieval, it is assumed that the information available in standard Web pages (the document base) is indexed using the semantic knowledge found in the SW. A key step in achieving this aim lies on linking the semantic space to the unstructured content space by means of the explicit annotation of documents with semantic data. In such dynamic and changing environment, **annotation must be done in a flexible and scalable way**. As we explain in the following sections, the solutions we are exploring in this work do not require hardwiring the links between Web pages and semantic markup. On the contrary these are created dynamically in such a way that the two information sources may remain decoupled.

In a similar way as traditional IR techniques base their ranking algorithms on keyword weighting, our approach relies on measuring the relevance of each individual association between semantic concepts and Web documents. In this way, not just the retrieval, but also the ranking of query answers can take advantage from the available semantic information.

Two different annotation methodologies are studied in this chapter. The fist one (described in section 6.2.1.1) uses Information Extraction (IE) methodologies in order to identify in the documents, words or groups of words that can potentially represent semantic entities (classes, properties, instances or literals). The second one (described in section 6.2.1.2) uses a more scalable approach based on statistical occurrences of semantic entities and their contextual semantic information. Both annotation procedures have been designed considering a set of common requirements:

- The semantic annotator identifies ontology entities (classes, properties, instances or literals) within the text documents, and generates the corresponding annotations. This is equivalent to a traditional IR indexing process where the indexing units are ontology entities (word senses) instead of plain keywords.
- The annotation processes carried out here do not aim to populate ontologies, but to identify already available semantic knowledge within the documents. In this way, the semantic information and the documents remain decoupled.
- Differently to other large scale annotation frameworks, our system has been designed to support annotation in open domain environments. Any document can be associated or linked to any ontology without any predefined restriction. The exploitation of massive amounts of

metadata and documents introduces scalability limitations. To address them, we propose the use of ontology indices, document indices and non-embedded annotations:

- *Generation of ontology indices*: We envision a scenario where the annotation module may need to interact with hundreds of KBs structured in tens of ontologies. To successfully manage such amount of information on real time, the ontologies and KBs are analyzed and stored into one or more inverted indices using Lucene. This index structures are part of the SW gateway module explained in chapter 7.
- *Generation of document indices*: A massive amount of unstructured content is currently available on the Web. To successfully manage such amount of information on real time, Web documents are pre-processed and stored in one or more inverted indices using Lucene.
- *Construction of the annotation database:* In contrast to systems where annotations are embedded in the ontologies or documents, our mechanism generates non-embedded annotations. These annotations are stored into a relational database, increasing the efficiency of the retrieval phase. For each annotation an entry is generated into the database. This entry contains the identifiers of the corresponding semantic entity (word sense) and document, as well as a weight indicating the degree of relevance of the semantic entity within the document. Weights are automatically computed using different techniques for the two annotations process presented (see sections 6.2.1.1 and 6.2.1.2 for further details). The relational model designed to store the above annotations is composed by the following tables:

    - *Annotations table*. This table stores the annotations, linking documents with ontology entities through weights.

Entity ID	Document ID	Weight
1829048176	3614522287	0.54
1829048179	3614522287	0.21

    - *Ontology Entity table*. This table stores index information about ontology entities. Each entity is identified by its ontology, its URI and its type (class, instances, property, literal), and has associated a set of text labels.

Entity ID	Entity URI	Entity type	Entity labels	Ontology ID
1829048176	0#Teide	instance	teide	45
1829048179	1#boat	class	boat, ship	46

    - *Document table*. This table stores information about the textual documents. Each document is identified by its URI and its repository or media source.

Document ID	Document URI	Repository ID
3614522287	24#CNN_D1	21
3614522289	24#CNN_D2	24

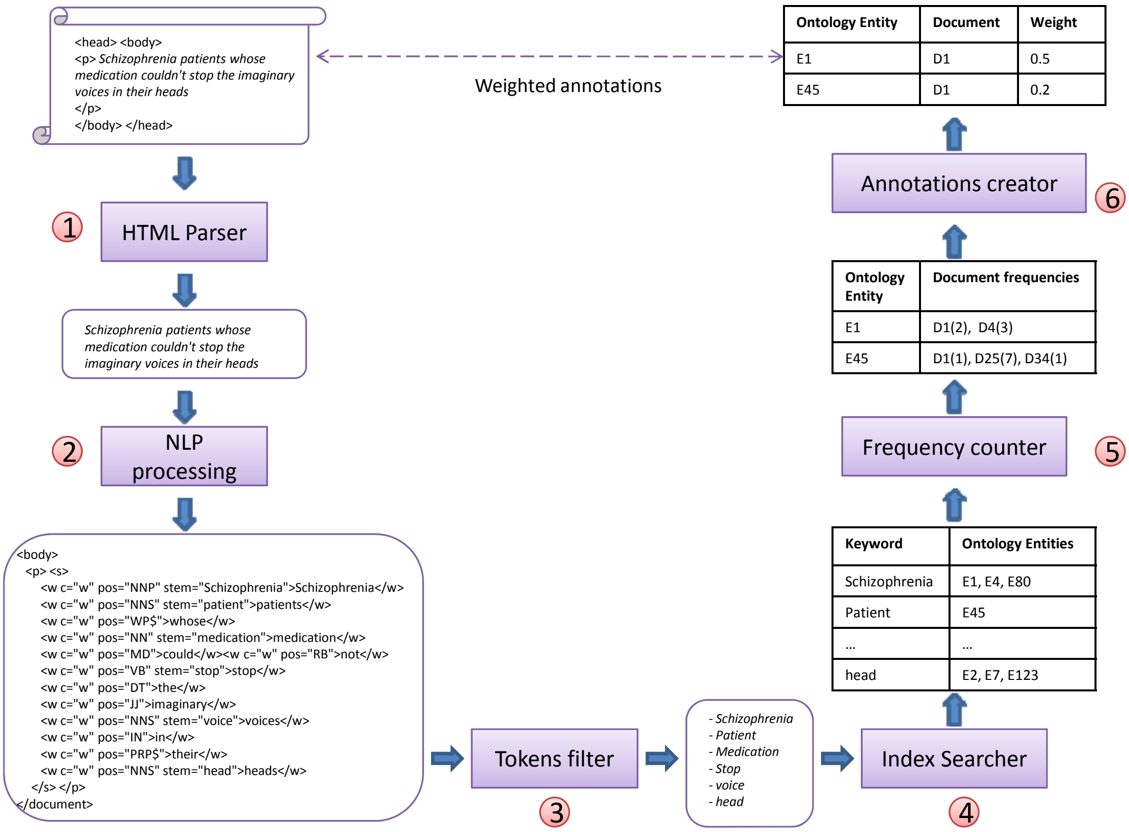
- *Prefix table*. This table optimizes the storage of namespaces in the database.

Prefix	Namespace
0	http://geography.com/spain/mountain
1	http://transports.net/watercraft
24	http://www.cnn.com/travel

The following sections present the implemented annotation processes. The first one analyzes textual documents using NLP techniques, extracts information from those documents and tries to map it with the semantic information stored in the ontologies and KBs. The second one works in the opposite direction. It analyzes the semantic information stored in ontologies and KBs and, considering each ontology entity and its semantic context, tries to identify the semantic entities within the textual documents to generate new annotations.

6.2.1.1 Annotation by NLP

Using a set of Natural Language Processing tools (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006), this annotation module analyzes the textual documents, removes stop words and extracts relevant (simple and compound) terms, categorized according to their Part of Speech (PoS): nouns, verbs, adjectives, adverbs, pronouns, prepositions etc. Then, terms are morphologically compared with the names of the semantic entities of the domain ontologies. The comparisons are done using an ontology index created with Lucene (see chapter 7), and according to fuzzy metrics based on the Levenshtein distance (Levenshtein V. I., 1966). For each term, if similarities above a certain threshold are found, the most similar semantic concepts are chosen and added as annotations of the news items. After all the annotations are created, a TF-IDF technique computes and assigns weights to them. Fig 6.2 shows a more detailed view of the annotation mechanism, which takes as input the HTML document to annotate, and the ontology indices, and returns as output new entries for the annotation database. The steps followed are:



```
<head> <body>
<p> Schizophrenia patients whose
medication couldn't stop the imaginary
voices in their heads
</p>
</body> </head>
```

Schizophrenia patients whose medication couldn't stop the imaginary voices in their heads

```
<body>
  <p> <s>
     <w c="w" pos="NNP" stem="Schizophrenia">Schizophrenia</w>
     <w c="w" pos="NNS" stem="patient">patients</w>
     <w c="w" pos="WP$">whose</w>
     <w c="w" pos="NN" stem="medication">medication</w>
     <w c="w" pos="MD">could</w><w c="w" pos="RB">not</w>
     <w c="w" pos="VB" stem="stop">stop</w>
     <w c="w" pos="DT">the</w>
     <w c="w" pos="JJ">imaginary</w>
     <w c="w" pos="NNS" stem="voice">voices</w>
     <w c="w" pos="IN">in</w>
     <w c="w" pos="PRP$">their</w>
     <w c="w" pos="NNS" stem="head">heads</w>
   </s> </p>
</document>
```

- *Schizophrenia*
- *Patient*
- *Medication*
- *Stop*
- *voice*
- *head*

Keyword	Ontology Entities
Schizophrenia	E1, E4, E80
Patient	E45
…	…
head	E2, E7, E123

Ontology Entity	Document frequencies
E1	D1(2), D4(3)
E45	D1(1), D25(7), D34(1)

Ontology Entity	Document	Weight
E1	D1	0.5
E45	D1	0.2

Fig 6.2 Document annotation by NLP

1) The textual Web documents are parsed to erase meaningless (in terms of essential content to be conveyed) HTML tags.

2) The remaining text is analyzed by the *Wraetlic* linguistic-processing tools to extract the PoS and the stem of each term.

3) The information provided by the linguistic analysis is used to filter the less meaningful terms (determinants, prepositions, etc.), and to identify those sets of terms that can operate as individual information units.

4) The filtered terms are searched in the ontology indices, obtaining the subset of semantic entities to annotate.

5) The annotations are weighted according to the semantic entity frequencies within individual documents and the whole collection.

6) The annotations are added to a relational database.

6.2.1.1.1 Text content processing

The Natural Language Processing of the annotation module is carried out by means of the *Wraetlic* linguistic-processing tools (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006), an XML suite for processing texts which performs the following tasks:

- **Segmentation**: the identification of lexical units in the texts. It is done by two components: a *tokenizer* which finds word boundaries, and a *sentence splitter* which locates the sentence boundaries. The tokenizer makes use of a list of regular expressions that define the different types of "tokens" appearing in the sentences, such as words, numbers or punctuation symbols. The sentence splitter analyzes the words followed by a dot to decide whether they are abbreviations or not, and uses this information to get the sentence boundaries.

- **Part-of-Speech (PoS) tagging**: the assignment of a PoS to each token. A *PoS tagger* labels each token with its corresponding PoS. *Wraetlic* tools utilize the PoS tags of the Penn Treebank corpus[45], and take into consideration the grammatical context of a word (i.e. its surrounding terms) to infer its PoS.

- **Morphological analysis**: the study of the inner structure of the words. For each token, a morphological analyzer identifies the root (stem), which contains the basic meaning of the word, and the bound morphemes (prefixes and suffixes), which vary the basic meaning, e.g., by pluralizing a noun (e.g., "parent" and "parents"), or by changing an adjective into a noun (e.g., "wide" and "width").

An example

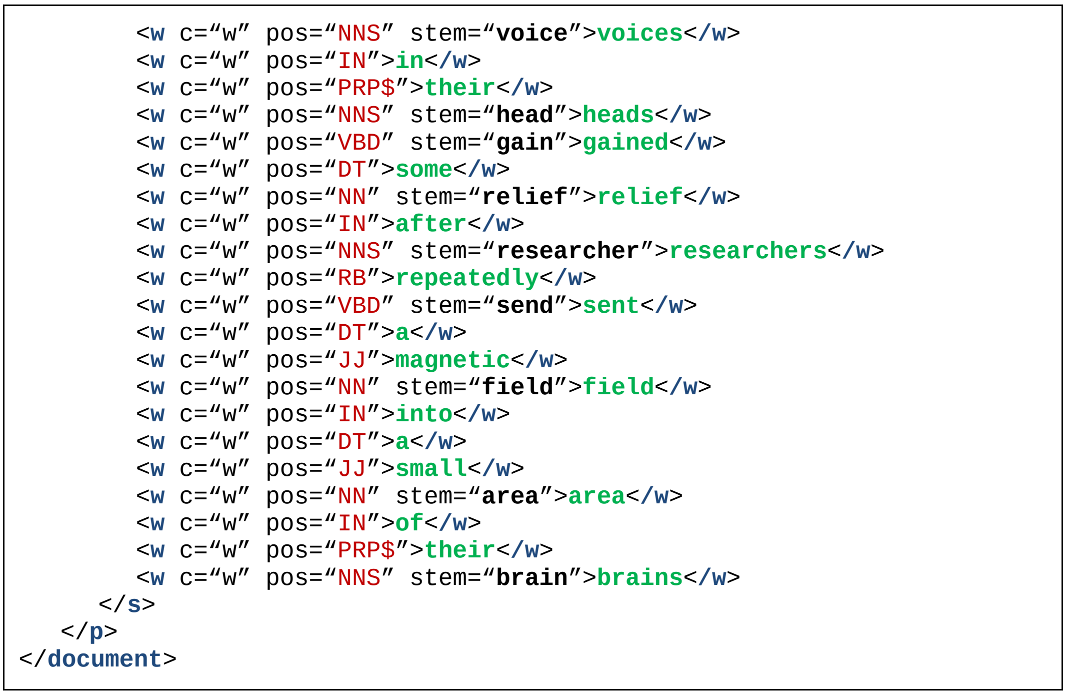
Suppose the following text as the content of a Web document to analyze (and annotate):

> Schizophrenia patients whose medication couldn't stop the imaginary voices in their heads gained some relief after researchers repeatedly sent a magnetic field into a small area of their brains.

The NLP performed by *Wraetlic* produces the following XML output:

```
<document>
   <p>
      <s>
         <w c="w" pos="NNP" stem="Schizophrenia">Schizophrenia</w>
         <w c="w" pos="NNS" stem="patient">patients</w>
         <w c="w" pos="WP$">whose</w>
         <w c="w" pos="NN" stem="medication">medication</w>
         <w c="w" pos="MD">could</w>
         <w c="w" pos="RB">not</w>
         <w c="w" pos="VB" stem="stop">stop</w>
         <w c="w" pos="DT">the</w>
         <w c="w" pos="JJ">imaginary</w>
```

[45] The Penn Treebank Project, http://www.cis.upenn.edu/~treebank



```
         <w c=“w” pos=“NNS” stem=“voice”>voices</w>
         <w c=“w” pos=“IN”>in</w>
         <w c=“w” pos=“PRP$”>their</w>
         <w c=“w” pos=“NNS” stem=“head”>heads</w>
         <w c=“w” pos=“VBD” stem=“gain”>gained</w>
         <w c=“w” pos=“DT”>some</w>
         <w c=“w” pos=“NN” stem=“relief”>relief</w>
         <w c=“w” pos=“IN”>after</w>
         <w c=“w” pos=“NNS” stem=“researcher”>researchers</w>
         <w c=“w” pos=“RB”>repeatedly</w>
         <w c=“w” pos=“VBD” stem=“send”>sent</w>
         <w c=“w” pos=“DT”>a</w>
         <w c=“w” pos=“JJ”>magnetic</w>
         <w c=“w” pos=“NN” stem=“field”>field</w>
         <w c=“w” pos=“IN”>into</w>
         <w c=“w” pos=“DT”>a</w>
         <w c=“w” pos=“JJ”>small</w>
         <w c=“w” pos=“NN” stem=“area”>area</w>
         <w c=“w” pos=“IN”>of</w>
         <w c=“w” pos=“PRP$”>their</w>
         <w c=“w” pos=“NNS” stem=“brain”>brains</w>
      </s>
   </p>
</document>
```

Fig 6.3 XML output provided by *Wraetlic*

As shown in Fig 6.3, the NLP tools parse the document, recognize its paragraphs, sentences, and tokens, and provide information about the PoS and the semantic stem of each token. This information will be used afterwards by the annotation module to discard meaningless tokens such as determinants, prepositions, etc., and to identify lexical structures (tokens or groups of tokens) which might potentially match with ontology entities.

6.2.1.1.2 Dealing with ambiguity

The use of a potentially unlimited number of domain ontologies ad KBs increases the uncertainty of the annotations, as more morphological similar concepts (with divergent semantic meanings) can be found. To address this limitation, we propose to exploit the PoS information provided by *Wraetlic* NLP tools in order to identify and discard those words that typically do not provide significant semantic information. Moreover, we attempt to group sets of words that can operate as individual semantic information units. Some examples of the considered word group patterns are the following:

- *Noun* + *noun*. E.g., “tea cup”.
- *Proper noun* + *proper noun*. E.g., “San Francisco”.
- *Proper noun* + *proper noun* + *proper noun*. E.g., “Federico García Lorca”.
- *Abbreviation* + *proper noun* + *proper noun*. E.g., “F. García Lorca”.
- *Abbreviation* + *abbreviation* + *proper noun*. E.g., “F. G. Lorca”.
- *Participle* + *preposition*. E.g., “located in”, “stored in”.
- *Modal verb* + *participle* + *preposition*. E.g., “is composed by”, “is generated with”.

6.2.1.1.3 Weighting annotations

As in the initial model (section 5.2.1.3) annotation weights are computed automatically by an adaptation of the TF-IDF algorithm, based on the frequency of the occurrences of each semantic entity within the document. The number of occurrences of a semantic entity in a document is primarily defined as the number of times any of its associate keywords appears in the document text. We realized in our first experiments that quite a number of occurrences were missed in practice, since the algorithm was not considering pronouns as semantic entity occurrences. To slightly overcome this limitation a modification of the algorithm was added to count pronoun occurrences in the scope of a sentence if a semantic entity was previously identified. This modification in the weighting algorithm has helped to better determine which semantic entities are really meaningful for a document. This fact does not help to increase the accuracy of the annotations or to add new ones, but it is even more important, since it enhances the accuracy of the annotation weights that will be later used during the ranking process. The weight of an annotation or the weight d_x of a semantic entity x for a document d is computed as:

$$d_x = \frac{freq_{x,d}}{max_y freq_{y,d}} \cdot log \frac{|D|}{n_x}$$

Where: $freq_{x,d}$ is the number of occurrences in d of the keywords attached to x, $max_y freq_{y,d}$ is the frequency of the most repeated semantic entity y in d, n_x is the number of documents annotated with x, and D is the set of all documents in the search space.

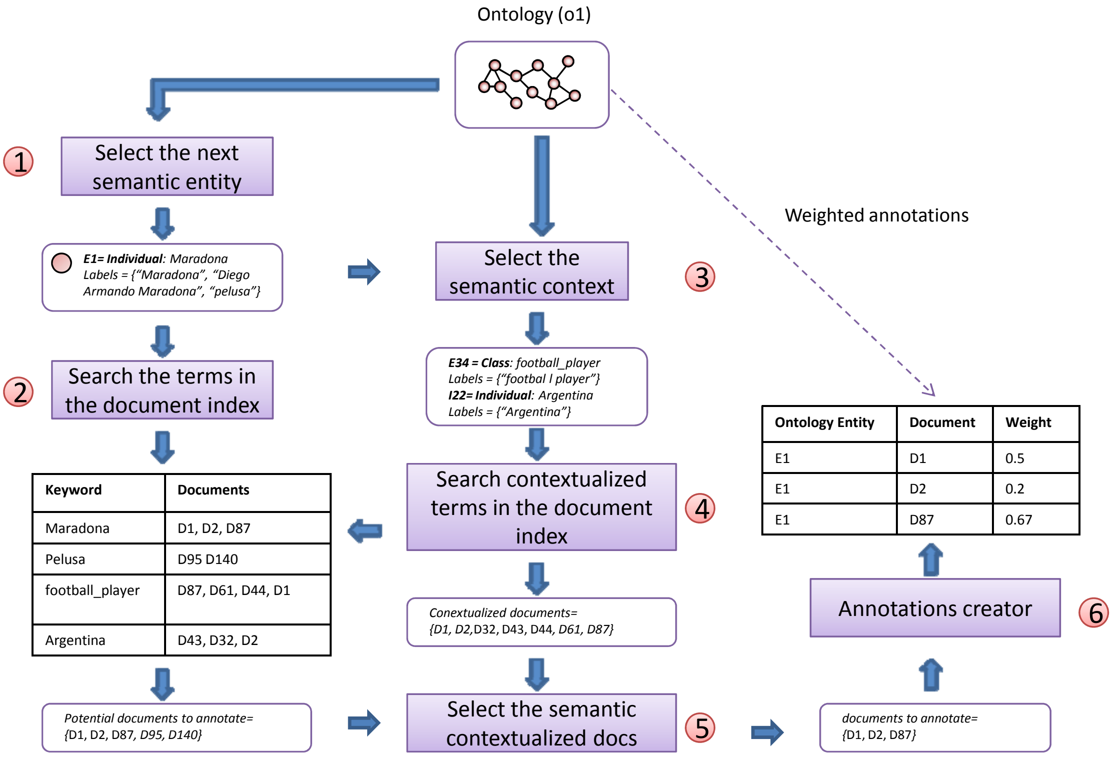
6.2.1.2 Annotation based on contextual semantic information

In the previous annotation mechanism (section 6.2.1.1) the documents were analyzed and the filtered terms were searched in the semantic entity index. In this one, in contrast, the semantic entities are the ones analyzed and searched in the document index (a standard keyword-based index generated prior to the annotation process). Inverting the direction of the annotation process, from semantic entities to documents, provides two important advantages: on one hand, the semantic information stored in the ontologies and KBs can be used as background knowledge to improve the accuracy of the annotations; on the other hand, the computational cost decreases because the textual documents have been indexed previously. This new annotation model constitutes a more scalable and widely applicable approach because it can potentially use any keyword-based document index, including the ones generated by companies like Google[46]or Yahoo[47].

The overall annotation process is shown in Fig 6.4, and consists of the following steps to be performed for every semantic entity in every ontology:

[46] www.google.com

[47] www.yahoo.com


Ontology (o1)
Weighted annotations
1
Select the next semantic entity
E1= Individual: Maradona
Labels = {"Maradona", "Diego Armando Maradona", "pelusa"}
2
Search the terms in the document index
Potential documents to annotate=
{D1, D2, D87, D95, D140}
3
Select the semantic context
E34 = Class: football_player
Labels = {"footbal l player"}
I22= Individual: Argentina
Labels = {"Argentina"}
4
Search contextualized terms in the document index
Conextualized documents=
{D1, D2,D32, D43, D44, D61, D87}
5
Select the semantic contextualized docs
documents to annotate=
{D1, D2, D87}
6
Annotations creator

Keyword	Documents
Maradona	D1, D2, D87
Pelusa	D95 D140
football_player	D87, D61, D44, D1
Argentina	D43, D32, D2

Ontology Entity	Document	Weight
E1	D1	0.5
E1	D2	0.2
E1	D87	0.67

Fig 6.4 Document annotation based on contextual semantic information.

1) *Load the information of a semantic entity*. Extract the textual representation of the selected semantic entity. Each semantic entity has one or more textual representations in the ontology. E.g., the individual entity describing the football player Maradona can be named as "Maradona", "Diego Armando Maradona", "Pelusa", etc. Here we assume that such lexical variants are present in the ontology as multiple values of the local name or rdfs:label property of the entity.

2) *Find the set of potential documents to annotate*. The textual representations of the semantic entity are then searched in the document index using standard search and ranking processes, in order to find the documents that may be associated with it. These documents simply contain the textual representation of the semantic entity, which does not necessarily imply that they contain its semantic meaning: they are candidates for annotation, to be considered by the following steps.

3) *Extract the semantic context of the entity*. The semantic meaning of a concept is determined by the set of concepts it is linked or related to in the domain ontology. To ensure that a semantic entity annotates the appropriate set of documents, we exploit the ontological relations to extract its context, that is, the set of entities directly linked in the ontology by an explicit relation. E.g., the semantic entity Maradona is related to the concepts Football player, Argentina, etc.

4) *Find the set of contextualized documents*. Thee textual representations of entities in the set of semantically related concepts, or semantic context, produced in the previous step, are then searched in the document index to extract the set of contextualized documents.

5) *Select the final list of documents to annotate*. We compute the intersection between the documents having textual representations of the semantic entity (extracted in step 2) and the set of documents having textual representations of the entities on its semantic context (extracted in step 4). Documents in this set are not just likely to contain the concept but also the contextual meaning of the concept in the ontology.

6) *Create the annotations*. A new entry or annotation is created for every document in the previous set. The annotation will have a weight indicating the degree of relevance of the entity within the document. The algorithm to calculate this annotation weight is explained in section 6.2.1.2.2.

6.2.1.2.1 Dealing with ambiguity

As we have shown in the previous section, to reduce the ambiguity of annotations we use, as background information, the context of the semantic entities. The context of a semantic entity is defined as the set of entities directly linked with it in the ontology by an explicit relation. Using this context, we are able to annotate entities with documents that contain the contextual meaning of the semantic entity in the ontology. We have empirically observed that, using this technique, we are gaining a lot of precision but, at the same time, we are losing an important quantity of annotations. One potential cause of this problem is the low density of relations appearing in the SW ontologies (D'Aquin, Gridinoc, Sabou, Angeletou, & Motta, 2007). In these cases, the ontologies do not have enough contextual information to identify the meaning of the entity in the document and, therefore, the annotation is not created. This trade-off between the quality and the quantity of annotations is an interesting research point for future work extensions.

6.2.1.2.2 Weighting annotations

In both, the original (section 5.2.1.3) and the NLP-based (section 6.2.1.1) annotation models, annotations weights are computed automatically by an adaptation of the TF-IDF algorithm based on the frequency of the occurrences of each semantic entity within the document.

In this new annotation approach the annotation weights are computed in the following way:

- The fusion methodology, described in chapter 8, is used on the ranked lists of documents obtained at steps 2 and 4 to produce a ranked list S of documents that are candidates for annotations and a ranked list C of contextualized documents for semantically related entities, respectively.
- A document d occurring in both, and hence selected for annotation by step 5, will be given weight $P\,S\_d + (1-P)\,C\_d$, where P is a constant used control the influence of the semantic contextualization. We empirically found that a value of $P = 0.6$ seems to work well in practice.

This new annotation weighting methodology is less dependent on potential changes in the ontologies and KBs. When a new semantic entity is added or modified, it is only necessary to recalculate its annotations and the annotations of the semantic entities directly linked to it in the ontology and KBs. However, it presents one main disadvantage: the use of document ranking scores to compute the annotation weights introduces a loss of accuracy. Lucene is based on the traditional TF-IDF weighting measure and therefore, it is able to compute keyword frequencies, but not semantic entity frequencies. With the aim to mitigate this problem, all the textual representations of a semantic entity are searched in the document index, and therefore all of them contribute to the weighting process.

6.2.2 Query processing

As introduced in the motivation of this chapter, two of the main barriers for bringing current semantic retrieval systems to the Web environment are: a) **usability** limitations (they generally require users to have prior knowledge about ontology-based query languages, or to manage complicate form-based interfaces to formulate their queries) and b) **heterogeneity** limitations (semantic search systems generally manage a set of predefined ontologies covering a limited set of domains and therefore they do not scale to heterogeneous document repositories such as the Web).

Aiming to overcome those limitations we have replaced our previous query processing module by an ontology-based QA system, PowerAqua (Lopez, Motta, & Uren, 2006) designed to exploit large scale, heterogeneous semantic data. Unlike its predecessor, AquaLog (Lopez, Pasin, & Motta, 2005), which derived an answer from a single ontology, PowerAqua performs question answering on an potentially unlimited number of ontologies. As such it is part of a new generation of SW tools which dynamically select, reuse and combine information drawn from multiple and heterogeneous ontologies (D'Aquin, et al., 2008). Note that this tool was kindly provided by the Knowledge Media Institute for our experimental purposes, but its research and development is not part of this thesis. A brief explanation of his system follows.

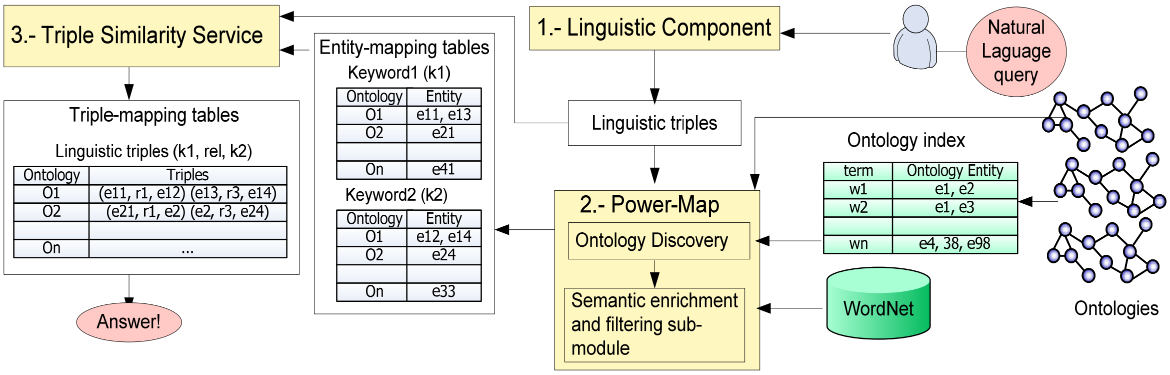


Fig 6.5 PowerAqua components in detail.

PowerAqua consists of three main components as shown in Figure 2. First, its **linguistic component** (detailed in (Lopez, Pasin, & Motta, 2005)) uses GATE (Cunningham, Maynard, Bontcheva, & Tablan, 2002) to translate a NL query into its linguistic triple form <*query term, relation, term*>, by identifying triple associations that relate terms together through verbs and prepositions. For instance, our example query "which are the members of the rock group Nirvana?", is translated to <*what-is*, *members, rock group nirvana*>. Second, **PowerMap** (Lopez, Sabou, & Motta, 2006) maps the terms of each linguistic triple to semantically relevant ontology entities (see Section 6.2.2.1). Finally, the PowerAqua **triple similarity service**, presented in (section 6.2.2.2), selects the ontological triples that best represent the user's query. An answer is then generated from these triples (e.g., as a list of instances that satisfy the input query).

6.2.2.1 The PowerMap algorithm

PowerMap, as detailed in (Lopez, Sabou, & Motta, 2006), is a hybrid knowledge-based matching algorithm comprising terminological and structural scheme matching techniques with the assistance of large scale ontological and lexical resources. PowerMap provides a mapping from linguistic terms to ontology entities. Given the linguistic triples identified by the linguistic component, PowerMap first identifies all the ontologies that are likely to describe the entities of these triples (i.e., those that contain syntactically similar entities). Then, it identifies the senses of the matches and excludes those that do not match the input linguistic terms semantically. The output of PowerMap is then a set of *Entity Mapping Tables* each corresponding to one linguistic term. Each table contains the ontology elements (drawn from different ontologies) to which the term was matched (See Table 6.1).

The PowerMap Ontology Discovery sub-module identifies, at run time, the set of ontologies likely to provide the information requested by the user. PowerMap is designed to work with an unlimited number of ontologies, thus taking advantage of the knowledge provided by the SW. To access this large amount of information in real-time, the tool has been integrated with a SW gateway (explained in chapter 7). In the implemented SW gateway, the semantic entities are indexed based on a mapping to a set of keywords that represent their meaning. These keywords are extracted, by default, from the entity's local name and its rdfs:label property and, optionally, from any other ontology property. These mappings allow the generation of an inverted index where each keyword may be associated to several semantic entities from different ontologies. To search the semantic information stored in the indices we make use of the advantages that Lucene provides for approximate searches. A second index level is also generated, which contains taxonomical information about each semantic entity. PowerAqua makes use of both levels of indexing to increase the speed of the mapping process, thus managing, the distributed semantic information in real time.

The ontology discovery module searches for approximate syntactic matches of the linguistic entities within the ontology indices. To broaden the search space, and bridge the gap between the user and ontology terminology, it uses not just the terms of the linguistic triple but also lexically related words obtained from WordNet. Moreover, it initiates a spreading activation search across ontologies to find additional terms that are lexically different from the original keywords. For example, synonyms are found through properties like owl:sameAs, while hypernyms and hyponyms are found by looking at the superclasses and subclasses of the ontology matches - an ontology about music can relate the term "group" as a hypernym of "band".

“Rock group nirvana”	
{}	[]
rock	
ATO[48]	foo:bar#Rock [class, exact, {Synset#1: rock, stone – material …}]
Music ontology	http://www.nets.ii.uam.es/music.owl#rock [instance, exact {Synset#2: rock_n_roll, rock_music}]
NALT[49]	http://agclass.nal.usda.gov/nalt/2006.xml#rock_gardens [class, partial. Synset#3: garden of rocks]
Nirvana	
Music ontology	http://www.nets.ii.uam.es/music.owl#nivana [instance, exact {Synset#1: group}]
SWETO[50]	http://lsdis.cs.uga.edu/proj/semdis/testbed/#SWEET_613112 (Nirvana Meratnia) [instance, partial] http://lsdis.cs.uga.edu/proj/semdis/testbed/#Region [class, hypernym (WN), Synset#2:location]

Table 6.1 Entity Mapping Tables example

Considering the previous mentioned example, the compound “rock group nirvana” does not produce any mappings as such, unless it is split into its parts. Consequently, the triple <which-is, members, rock group nirvana> is then split into the set of triples <which-is, members, nirvana>, <rock, ?, nirvana> and <group, ? nirvana>. As shown in Table 6.1, which presents the Entity Matching Table, the term “nirvana” has an approximate equivalent match with the instance of “researcher”, labeled “Nirvana Meratnia”, in the SWETO ontology, and an exact match with the instance of “group” labeled “nirvana” in the music ontology, among others. The term “rock” has two exact matches in the ATO (parent “substance”) and the music ontology (parent “specific-genre”) and a partial match with “rock_gardens” in the NALT ontology (parent “gardens”). For the sake of simplicity, we omit from the table the term “member”, which produce a large number of mappings.

Once, the set of possible syntactic mappings have been identified, the **PowerMap semantic enrichment and filtering sub-module** determines the sense of the identified entities and, when enough information is available, discards those matches that are semantically inappropriate. The semantic similarity between the triple terms and the concepts from distinct ontologies (in the case of instances it takes the class they belong to) is computed by taking into account their meaning as given by their place in the hierarchy of the ontology, through the use of a WordNet based methodology. In the case of the matches identified for “rock”, this step determines that the ATO match has the #*material, rock, stone* synset, that the music ontology entity belongs to the #*a genre of popular music* synset and that the NAL match represents the #*a garden featuring rocks* synset. Then, because the set of all possible sysnsets of rock in WordNet do not contain the NALT synset, this match is discarded. In

[48] http://reliant.teknowledge.com/DAML/ATO_Ontology.owl
[49] http://www.nal.usda.gov/fnic/foodcomp/Data
[50] http://lsdis.cs.uga.edu/projects/semdis/SWETO

cases when such false mappings cannot be identified due to lack of information, a more in depth filtering is performed by considering the context of the query and the ontology semantics (taxonomy and relationships) in the next step. This methodology, evaluated and discussed in (Gracia, Lopez, D'Aquin, Sabou, Motta, & Mena, 2007), provides good precision for filtering meaningful mappings, and has a low negative impact on recall (discarding relevant elements).

6.2.2.2 The triple similarity service

The Triple Similarity Service is invoked after all linguistic terminology has been meaningfully mapped at the element level. From these individual mappings spread over several ontologies (the *Entity Mapping Tables*), ontology relations are analyzed and the ontology compliant triples that semantically link those mappings and best represent the user query, are created. This step will return a small set of ontologies that jointly cover the user's query and contain enough information to deduce the answer to the question. The output is represented as *Triple Mapping Tables* that relate each linguistic triple to all the equivalent ontological triples obtained from different ontologies at the schema-level (Fig 6.5). For example, Table 6.2 shows the Triple Mapping Tables generated for two NL queries, identifying the source ontologies and the mechanism by which each term of the triple was matched. Finally, all the ontology triples that are related to actual instances, and therefore can be used to generate an answer, are selected, giving priority to the ontologies that contain the most matches to the individual terms of a linguistic triple. In other words, the algorithm selects the ontologies with the best coverage of a triple.

Then, the Relation Similarity Service (RSS) inspects each selected ontology and identifies the relations between the individual entities covered by an ontology in such a way that these relations are appropriate translations of the linguistic triples. As a result, a linguistic triple can be mapped into one or more ontology triples, each one belonging to the same or different ontologies, and those may represent complete alternative translations of the linguistic triple, or partial translations to be joined.

Which singers play rock? : <singers, play, rock>	
music ontology	<musicians (class-hypernym), has-members (property-ad hoc), rock (instance-exact)>
Find me all cities of Spain. : <*what-is*, cities, Spain>	
fao-agrovoc[51]	<city (class-synonym), generic-location (property-ad hoc), Spain (instance-exact)>
SWETO	<city(class-synonym), attribute-country (prop-ad hoc), spain (literal-exact)
Show me rock albums: <what-is, ?, rock albums>	
Music ontology	<album (class–synonym), has-albums (prop-approx.), rock (instance –exact)> <group (class–ad hoc), has-genre (prop-ad hoc), rock >

Table 6.2 Triple Mapping Tables example

[51] http://www.fao.org/agrovoc

The Relation Similarity Service: here we present a few representative examples to illustrate the algorithm:

- Case 1: Consider the NL query "Show me rock albums", which is translated to the linguistic triple <rock, ?, albums>, and let's assume that no ontologies were found to contain a match for the entire relation. However, there exist ontologies which contain matches for the individual entities of the triple (i.e., rock and albums). This is a rather typical case, either because the linguistic relation is implicit, as in this example, or because the ontology relation has a label that is difficult to detect by syntactic techniques, or because the relation is mapped to an ontology class (e.g., in "which are the cities of Spain", the relation "are the cities of" is mapped to an ontology class labeled "city"). Therefore the problem becomes one of finding ad-hoc relations that link the two terms, i.e. <city, has-attribute-country, Spain>. If no ad-hoc relations are found then IS-A relations between the arguments are inspected. If such relations are not found either, then the algorithm investigates the existence of indirect relations through one mediating concept between the arguments. In this case, our query is translated into: <albums, has-albums, rock> <group, has-genre, rock>.
- Case 2: If unlike the previous case the algorithm identifies a set of candidate relations (e.g., in our illustrative example <what-is, members, nirvana>), then matching and joining of triples is controlled by the domain and range information of the relations and the mapped ontology elements (e.g., the resulting ontology triple is <musicians, has-members, nirvana>). In many cases, in order to interpret a linguistic triple within one ontology, studying the ontology "neighborhood" of the potential matches may lead not only to linking the mapped terms between themselves but also to finding possible matches for those triple elements that could not be mapped through the previous syntactic and semantic matching stages (e.g., "musicians" is the term that completes the triple <musician, has-member, nirvana>).
- Case 3. If there are candidate matches for both the arguments of the triple and the relation between them, but none of the corresponding ontology triples can produce an answer, then the RSS ignores the relation name and initiates a search for ontology triples between the arguments only. The rationale behind this is that a relation's meaning is mostly given by the type of its domain and its range rather than by its name. Similarly, if the ontologies with better coverage do not produce any valid triples, or the retrieved set of triples do not produce any answer, the search is extended to ontologies with lower coverage.

Following this algorithm the query "*which are the members of the rock group Nirvana*?" produces the triples <musicians, has-members, nirvana> <nirvana, has-genre, rock> <nirvana, *is-a*, group>, from which the following answers, in the form of a list of instances, are obtained: Dan_peters, Dave_grohl, and Kurt_cobain, among others. Note that other mappings of nirvana, rock and group in other ontologies have been discarded as they did not produce any relevant ontological triples.

As we have described in this section, the integration of PowerAqua as the query processing module of our semantic retrieval system brings two main advantages to our research: a) the possibility to process natural language queries, increasing the level of usability of our application without losing expressivity and, b) the ability to retrieve answers from a massive number of ontologies at a time, therefore dealing with heterogeneity limitations.

6.2.3 Searching and ranking

The semantic document-retrieval and ranking approach presented here remains from our initial design (see section 5.2.3), except for the way in which the query vector is constructed. As we explained before, the retrieval and ranking algorithm is based on an adaptation of the traditional vector-space IR model (see section 2.3.2) where documents and queries are represented as weighted vectors. Fig 6.6 illustrates our proposed adaptation of the vector-space model that replaces the traditional keyword query and document vectors by a semantic query and document vectors. The query vector represents the importance of each semantic entity in the information need expressed by the user, while the document vector represents the relevance of each semantic entity within the document.

The construction of the document vector remains from our previous model, but the construction of the query vector has been adapted to manage the degree of uncertainty of the answers retrieved by PowerAqua. Note that, in the original model, the input was a formal SPARQL query. This query was executed against the KB returning as answer a list of instance tuples in a purely Boolean step (i.e. based on an exact match). To introduce the importance of the different concepts in the information need expressed by the query, or its discriminating power for discerning relevant from irrelevant documents, the variables in the SELECT clause of the SPARQL query were weighted.

Using PowerAqua as query processing module already introduces a degree of uncertainty in the retrieved answers: a) the ontology discovery module searches for *approximate syntactic matches* in order to find the ontologies that can potentially answer the user′s query, b) the semantic enrichment and filtering sub-module *disambiguates the sense of the identified entities* using as background knowledge the available semantic information and c) the relation similarity service *maps the constructed linguistic triple into one or more ontology triples*, each one belonging to the same or different ontologies.

At the time of carrying out the experiments of this thesis, the degree of uncertainty of the retrieved answers was not measured by PowerAqua, and no ranking or score was provided. Therefore, all the semantic entities were retrieved by the query module with the same degree of relevance. As a simple approach we decided to introduce a query weighting measure considering the set of semantic entities retrieved for each detected query condition. For example, if the user asks for "symptoms and treatments of Parkinson disease" PowerAqua is able to retrieve as answer a set of individual symptoms and a set of individual treatments. Considering that $SEci$ is the set of semantic entities retrieved for the query condition i, the weight of each retrieved semantic entity in the query vector is computed as $1/|SEci|$. The intuition behind this measure is that those query variables for which less ontology entities have been retrieved are more likely to be representative of the user information needs, and therefore they should be considered more important. Even though the performance of this measure has not been individually tested; several experiments have empirically proved that incorporation of this measure improves the precision of the document retrieval algorithm. Explicit information could be used to measure the relevance of each individual entity, such as the number of potential ontologies to answer the query, the number of query conditions answered by each ontology, the number of entities retrieved for each ontology and each query condition, etc. As a future work extension it will be interesting to evaluate the degree of uncertainty of the retrieved entities, as well as the set of features more likely to be used for its computation. Following this line, another possible extension may be the introduction of other ontology entity relevance measures such as the ones developed by (Stojanovic, Studer, & Stojanovic, 2003).

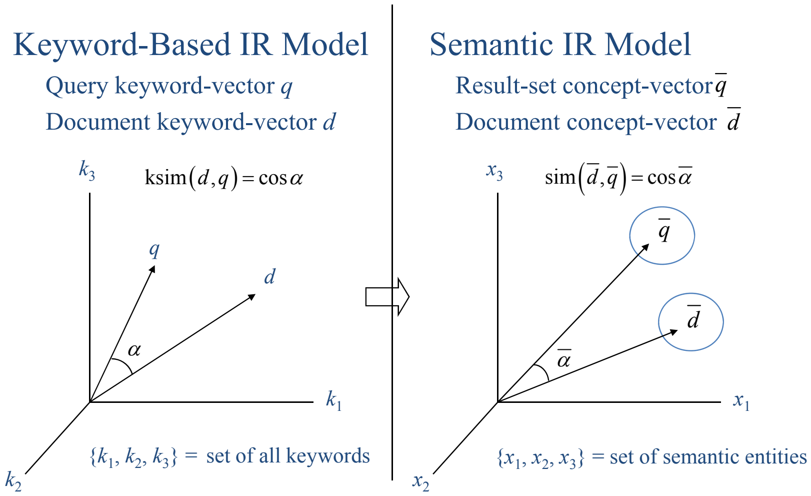

Keyword-Based IR Model
Query keyword-vector q
Document keyword-vector d
k_3
$\text{ksim}(d,q)=\cos\alpha$
q
d
α
k_1
k_2
$\{k_1, k_2, k_3\}$ = set of all keywords
Semantic IR Model
Result-set concept-vector $\bar{q}$
Document concept-vector $\bar{d}$
x_3
$\text{sim}(\bar{d},\bar{q})=\cos\bar{\alpha}$
$\bar{q}$
$\bar{d}$
$\bar{\alpha}$
x_1
x_2
$\{x_1, x_2, x_3\}$ = set of semantic entities

Fig 6.6 Adaptation of the vector-space model.

6.3 Evaluation

As described in section 5.4, in contrast to traditional IR communities, where evaluation using standardized techniques, such as those prescribed by the TREC annual competitions, has been common for decades, the SW community is still a long way from defining standard evaluation benchmarks to judge the quality of the current semantic retrieval methods. Current approaches for SW technologies evaluation are based on user-centered methods (Sure & Iosif, 2002) (McCool, Cowell, & Thurman, 2005) (Todorov & Schandl, 2008) and therefore they tend to be high-cost, non-scalable and difficult to repeat, especially at a Web scale.

Nonetheless, we want to test our system systematically and as rigorously as we could. To do so we had no choice but to build our own benchmark. We required a text collection, a set of queries and corresponding document judgments, ontologies that cover the query topics and KBs that populate the ontologies, preferably using a source independent of the text collection.

6.3.1 Evaluation benchmark

- **The Document Collection and Queries**: We decided to construct a benchmark taking the TREC 9 and TREC 2001 test corpora as a starting point, because this provides us with an independently produced set of queries and document judgments. The IR collection we took as basis comprises 10 GB of Web documents known as the TREC WT10G collection, 100 queries, corresponding to real user logs requests, and the list of document judgments related to each query. These judgments allow the quality of the information retrieval techniques to be calculated using standard precision and recall metrics.

- **The Ontologies**: To evaluate semantic retrieval systems we also need ontologies. However, as the SW is still sparse and incomplete (Sabou, Gracia, Angeletou, D'Anquin, & Motta, 2007), many of the query topics associated with WT10G are not yet covered by it. Indeed, we have only found ontologies covering around 20% of the query topics. In the remaining cases, ontology-based technologies cannot be used to enhance traditional search methodologies, and the system just relies on keyword-based search techniques to retrieve and rank Web documents. We have used 40 public ontologies which potentially cover a subset of the TREC domains and queries. These ontologies are grouped in 370 files comprising 400MB of RDF, OWL and DAML. In addition to the 40 selected ontologies, our experiments also access another 100 repositories (2GB of RDF and OWL) stored and indexed with the SW gateway indexing structures explained in chapter 7.
- **The Knowledge Bases**: Sparseness is an even bigger problem for KBs than for ontologies. Current publicly available ontologies contain significant structural information in the form of classes and relations. However, most of these ontologies are not populated or barely populated. As a result the available KBs are still not enough to perform a large-scale semantic retrieval testing. To overcome this limitation and provide a medium-scale test experimentation of our algorithms, some of the 40 selected ontologies have been semi-automatically populated from an independent information source: Wikipedia (the population approach is discussed in detail in section 6.3.1.1). Wikipedia is a public encyclopedia comprising knowledge about a wide variety of topics. In this way, we endeavor to show how semantic information publicly available on the Web can be applied to enhance keyword search over unstructured contents.

6.3.1.1 Populating ontologies using Wikipedia

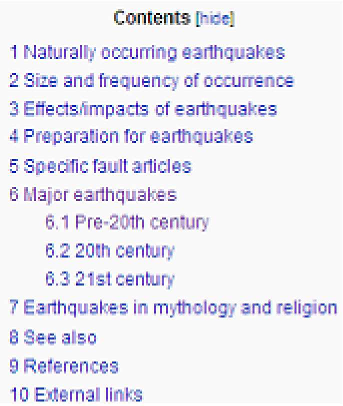
Here we present a simple semi-automatic ontology-population mechanism that can be, in principle, further improved with more sophisticated ontology population techniques, but this is out of the extent of this research. The algorithm here comprises two main functionalities: 1) populating an ontology class with new individuals; e.g., populating the class Earthquake with individuals such as 2007 Peru earthquake, 2007 Guatemala Earthquake, etc., and 2) extracting ontology relations for a specific ontology individual, e.g., extract relations for the individual Jennifer Aniston, such as the set of films she has acted in, etc.

Basically the algorithm comprises 5 steps:

1) The user selects the class of individuals he wants to populate or expand with new relations.
2) The system extracts the textual form of this concept: either from the localName, from the standard property rdf:label or from the non-standard but common ontology property (name or hasName).
3) The system looks for the textual form of the concept in Wikipedia.
4) The Contents section or index of the Wikipedia entry (see Fig 6.7) is used to generate new classes and/or relations for the index sections which point to a table (see Fig 6.9) or a list

(see Fig 6.8) that can be used to populate the ontology. Note that new classes and relations are created if we can not previously find a mapping in the ontology.

5) The classes selected by the user and the new generated classes (in step 4) are populated with the Wikipedia lists and/or tables. To generate a new individual from list entries we take as individual name the list row until we find a punctuation symbol and the rest of the content as part of the individual rdfs:comment property. To generate a new individual from a table we first create a new class with a set of properties corresponding to the table columns. For each row of the table we create a new individual of this class.



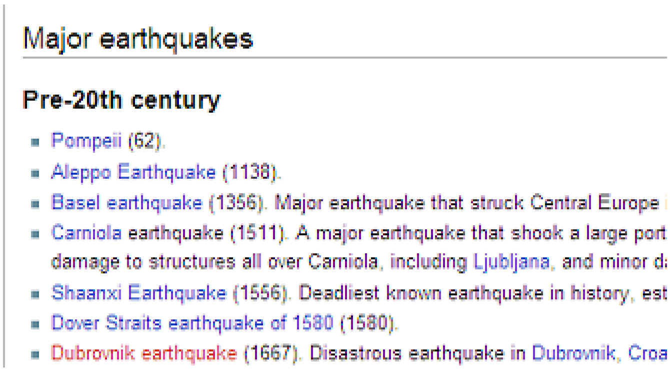
Contents [hide]

1 Naturally occuring earthquakes
2 Size and frequency of occurrence
3 Effects/impacts of earthquakes
4 Preparation for earthquakes
5 Specific fault articles
6 Major earthquakes
- 6.1 Pre-20th century
- 6.2 20th century
- 6.3 21st century

7 Earthquakes in mythology and religion
8 See also
9 References
10 External links

Fig 6.7 Example of Wikipedia contents table

E.g., let’s look in Wikipedia for the concept Earthquake: after analyzing the sections pointed to by the contents table shown in Fig 6.7 the system detects that sections 4, 5 and 6 contain potential lists to populate and extend the concept and asks the user to select the ones he wants to exploit. In this case we have chosen section number 6. The system then analyzes section number 6 in the contents to generate new classes and properties. First it detects a mapping between “MajorEarthquakes” and “Earthquakes”, so it does not create a new class but uses the one in the ontology. For this class the system adds three new subclasses “pre-20 century”, “20th century” and “21st century”. For each subclass the system creates the corresponding instances taking into account the Wikipedia lists. The list showed in Fig 6.8 contains the potential instances for the “Pre-20 century” subclass. After analyzing the first entry of the list the system creates the individual Pompeii and adds the rest of the information “(62)” to the its rdfs:comment property.



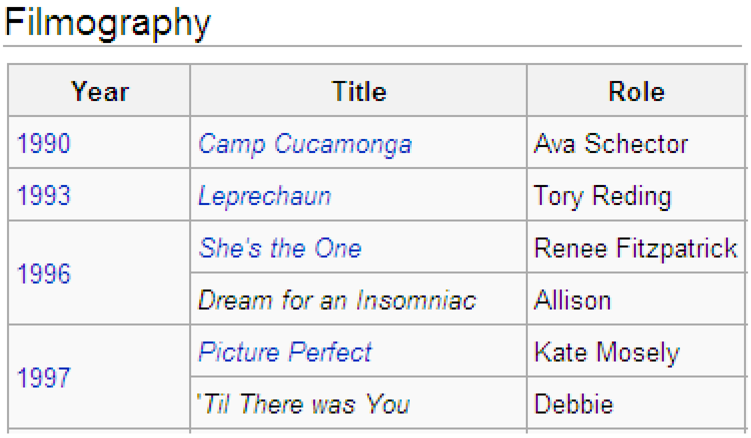
Major earthquakes

Pre-20th century

- Pompeii (62).
- Aleppo Earthquake (1138).
- Basel earthquake (1356). Major earthquake that struck Central Europe
- Carniola earthquake (1511). A major earthquake that shook a large port damage to structures all over Carniola, including Ljubljana, and minor d:
- Shaanxi Earthquake (1556). Deadliest known earthquake in history, est
- Dover Straits earthquake of 1580 (1580).
- Dubrovnik earthquake (1667). Disastrous earthquake in Dubrovnik, Croa

Fig 6.8 Example of Wikipedia list

With the tables the population process is slightly different. E.g, the table shown in Fig 6.9 is extracted from the Filmography section of the Jennifer Aniston Wikipedia entrance. For this section the algorithm generates the class "Filmography" with properties: "has year", "has title" and "has role". It also generates the property "hasFilmography" to link the individual "JenniferAniston" with the new "Filmography" individuals created from each row of the table.



Filmography

Year	Title	Role
1990	*Camp Cucamonga*	Ava Schector
1993	*Leprechaun*	Tory Reding
1996	*She's the One*	Renee Fitzpatrick
	Dream for an Insomniac	Allison
1997	*Picture Perfect*	Kate Mosely
	'Til There was You	Debbie

Fig 6.9 Example of Wikipedia table

This algorithm is supervised by the user. He identifies the ontology classes to populate, or the ontology instances to extend. He selects from the suggested Wikipedia sections the ones to be used, and he can modify the automatically generated names of classes and properties during the population process. With this algorithm we have generated around 20.000 triples distributed along the 40 preselected ontologies. As we said before, this new data added to the KBs have not been extracted from the TREC documents, but from Wikipedia, which maintains the independence assumption for our experiments between the SW data and the unstructured information to be retrieved. It is not our aim to research ontology population methods, but to take advantage of simple methodologies to show how semantic information publicly available on the Web can be applied to enhance keyword search over unstructured documents. Better automatic ontology-population methods can be therefore used to extend the publicity available semantic content, which will improve the quality of semantic retrieval approaches.

6.3.1.2 Adapting the TREC queries

In selecting the TREC queries we could use in our evaluation, we had two practical constraints. First, the queries must be able to be formulated in a way suitable for QA systems to be processed by PoweraAqua, this means queries like "*discuss the financial aspects of retirement planning*" (topic 514) can not be tackled. Second, ontologies must be available for the domain. As discussed above, the second point is a serious constraint. In the end, we considered 20 queries.

As we can see in Table 6.3, the original TREC queries are described by: a) a title, which is the original user query extracted from users' logs, b) a description, which can be considered the Natural

Language interpretation of the query, and c) the narrative, which explains in more detail the relevant information that the user is looking for. We added, for the queries we used: d) a detailed request, suitable for a question answering approach, e) notes on available ontologies. The complete list of our selection of TREC topics and its adaptation is available in Appendix B.

The final evaluation benchmark comprises: a) The TREC WT10G collection of documents; b) 20 queries and their corresponding judgments extracted from the TREC 9 and TREC 2001 competitions; c) 40 public ontologies, some of them populated from Wikipedia, covering the domains of the 20 selected queries and d) an 2GB of extra amount of public available Semantic Data, provided by the SW gateway integrated into our system.

Num	Number: 494
Title	nirvana
Desc	Find information on members of the rock group Nirvana.
Narr	Descriptions of members' behavior at various concerts and their performing style is relevant. Information on who wrote certain songs or a band member's role in producing a song is relevant. Biographical information on members is also relevant.

Table 6.3 Example of TREC query

6.3.2 Experimental conditions

The experiments were designed to compare the results obtained by four different search approaches at a Web scale:

- **Keyword search**: a conventional keyword-based retrieval approach, using the Jakarta Lucene library[52].
- **Semantic search:** our complete semantic retrieval system, including the query processing performed by Power Aqua, the semantic retrieval and ranking subsystem, and the rank fusion methodology reported in chapter 8.
- **Best TREC automatic search**: the approach used by the best TREC search engine that uses as query just the title section.
- **Best TREC manual search**: the approach used by the best TREC search engine which manually generates the queries using information from the title, the description and the narrative.

We have decided to include in our evaluation the results obtained by the best TREC search engines (title-only and manual) of TREC 9 and TREC 2001 competitions. However, there are several concerns with the TREC benchmark that should be considered:

[52] http://lucene.apache.org

- *The judgements*: the judgments for each query of TREC 9 and TREC 2001 competitions are obtained using the pooling method described in section 2.4.2.3. In this methodology, methods that did not contribute to the pools might retrieve unjudged documents that are assume to be non-relevant, which, as described in later studies (Voorhees E. , 2001) leads to their evaluation scores being deflated relative to the methods that did contribute.

- *The queries*: the queries selected for TREC 9 and TREC 2001 are extracted from real Web search engine logs. That means that, the queries are generated in a suitable way for traditional keyword-based search engines, and therefore lack of expressivity in terms of relationships and query conditions.

- *The query construction*: in TREC 9 and TREC 2001 different evaluation categories are considered depending on how the input queries are formulated: a) using just the title b) automatically constructing the query from the title and the description and c) manually constructing the query using the title, the description and the narrative. A better performance is expected from approaches which manually constructed the queries than from those that use just the title because they add a significant amount of additional information to the query. As it is explained in the previous section, our approach manually modifies the queries to formulate them in a way suitable for QA systems, but minimizing the amount of information added to the query. Therefore, the comparison of the TREC manual approach with the other three approaches is not totally fair.

6.3.3 Results

Table 6.4 and Table 6.5 contain the results of our performed evaluation using the 20 TREC topics and two standard IR evaluation metrics: mean average precision (MAP) and precision at 10 (P@10) for each of the approaches evaluated. The first metric shows the overall performance of the system in terms of precision, recall and ranking. The second one shows how the system works in terms of precision for the top-10 results, which are the ones most likely to be seen by the user.

Numbers in bold correspond to maximal results for the current topic under the current metric, excluding the Best TREC manual approach, which outperforms the others significantly by both metrics likely because of the way the query is constructed: introducing information from the title, the description and the narrative. The other three methodologies construct the query either using just the title, in the case of the best TREC automatic approach, or using the title, and some parts of the description, in the case of Lucene and our semantic retrieval engine. For this reason, **we will exclude Best TREC manual for the rest of our analysis**.

Note also that, for this experiment, the semantic retrieval approach uses the annotation process described in section 6.2.1.2.

Topic	Semantic	Lucene	TREC automatic	TREC manual
451	0.42	0.29	**0.58**	0.54
452	0.04	0.03	**0.2**	0.33
454	0.26	0.26	**0.56**	0.48
457	**0.05**	0	**0.12**	0.22

465	**0.13**	0	0	0.61
467	0.1	**0.12**	0.09	0.21
476	0.13	0.28	**0.41**	0.52
484	**0.19**	0.12	0.05	0.36
489	**0.09**	**0.11**	0.06	0.41
491	**0.08**	**0.08**	0	0.7
494	0.41	0.22	**0.57**	0.57
504	0.13	0.08	**0.38**	0.64
508	**0.15**	0.03	0.06	0.1
511	0.07	0.15	**0.23**	0.15
512	0.25	0.12	**0.3**	0.28
513	0.08	0.06	**0.12**	0.11
516	**0.07**	0.03	**0.07**	0.74
523	**0.29**	0	0.23	0.29
524	**0.11**	0	0.01	0.22
526	**0.09**	0.06	0.07	0.2
Mean	0.16	0.1	**0.2**	0.38

Table 6.4 Quality of results by MAP

Topic	Semantic retrieval	Lucene	TREC automatic	TREC manual
451	0.7	0.5	**0.9**	0.8
452	0.2	0.2	**0.3**	0.9
454	0.8	0.8	**0.9**	0.8
457	0.1	0	**0.1**	0.8
465	**0.3**	0	0	0.9
467	**0.4**	**0.4**	0.3	0.8
476	**0.5**	0.3	0.1	1
484	0.2	**0.3**	0	0.3
489	0.2	0	0.1	0.4
491	0.2	**0.3**	0	0.9
494	0.9	0.8	**1**	1
504	0.2	0.2	**0.5**	1
508	**0.5**	0.1	0.3	0.3
511	0.4	0.5	**0.7**	0.2
512	**0.4**	0.2	0.3	0.3
513	0.1	**0.4**	0	0.4
516	**0.1**	0	0	0.9
523	**0.9**	0	0.4	0.9
524	**0.2**	0	0	0.4
526	**0.1**	0	0	0.5
Mean	**0.37**	0.25	0.3	0.68

Table 6.5 Quality of results by P@10

	Lucene		TREC automatic	
Topic	map	P@10	map	P@10
451	**0.7**	**0.7**	1.4	1.3
452	**0.8**	**1.0**	5.1	1.5

454	**1.0**	**1.0**	2.1	1.1
457	**0.0**	**0.0**	2.4	**1.0**
465	**0.0**	**0.0**	**0.0**	**0.0**
467	1.3	**1.0**	**0.9**	**0.8**
476	2.2	**0.6**	3.3	**0.2**
484	**0.6**	1.5	**0.2**	**0.0**
489	1.2	**0.0**	**0.6**	**0.5**
491	**1.0**	1.5	**0.0**	**0.0**
494	**0.5**	**0.9**	1.4	1.1
504	**0.6**	**1.0**	2.8	2.5
508	**0.2**	**0.2**	**0.4**	**0.6**
511	2.1	1.3	3.1	1.8
512	**0.5**	**0.5**	1.2	**0.8**
513	**0.8**	4.0	1.5	**0.0**
516	**0.5**	**0.0**	**0.9**	**0.0**
523	**0.0**	**0.0**	**0.8**	**0.4**
524	**0.0**	**0.0**	**0.1**	**0.0**
526	**0.7**	**0.0**	**0.8**	**0.0**
Mean	**0.7**	**0.8**	1.5	**0.7**

Table 6.6 Comparative of semantic retrieval vs. Lucene and vs. Best TREC automatic

As we can see in Table 6.5, **by P@10, the semantic retrieval outperforms the other two approaches**, providing maximal quality for 55% of the queries and it is only outperformed by both Lucene and TREC semantic in one query (511). Semantic retrieval provides better results than Lucene for 60% of the queries and equal for another 20%. Compared to the best TREC automatic engine, our approach excels at 65% of the queries and produces comparable results at 5%. Indeed, the highest average value for this metric is obtained by semantic search.

The results by MAP are interesting. In those, there is no clear winner. While the average rating for Best TREC automatic is greater than that for semantic retrieval, semantic retrieval outperforms TREC automatic in 50% of the queries and Lucene in 75%.

Table 6.6 compares the results obtained by the three approaches. The numbers indicates the ratio of the quality of the results retrieved by the corresponding engine divided by the quality of the results retrieved by the semantic retrieval (i.e., a value greater than 1 indicates that semantic retrieval was outperformed). Values in bold are those were the quality of the results for a given engine was less than or equal to that of the results by semantic search.

We hypothesize that the quality of the results retrieved by semantic retrieval and it's measurement under MAP may be being adversely affected by the two following factors:

- More than half of the documents retrieved by the semantic retrieval approach have not been evaluated in the TREC collection. Therefore, our metrics marked them as irrelevant, when, in fact, some of them are relevant. In section 6.3.3.1 we study the impact of this effect and we manually evaluate some results to analyze how the semantic retrieval approach would perform if all documents had been evaluated.

- The annotation process used for the semantic retrieval approach is very restrictive (see section 6.2.1.2). In order to increase the accuracy of annotations, an annotation is generated when a document contains not just a concept but also its semantic context. If the concept appears in the document with a semantic context not reflected in its ontology, the annotation is not generated. Thus, the process discards possible correct annotations. The impact of this effect is studied in section 6.3.3.2. A new experiment is performed with a different annotation model with the aim to increase the amount of annotations while maintaining their overall quality.

The aforementioned sections also explain why these factors affect the MAP measurements much more than the P@10 measurements.

Another three relevant conclusions can be extracted from this evaluation:

- **For some queries for which the keyword search (Lucene) approach finds no relevant documents, the semantic search does**. This is the case of queries 457 (*Chevrolet trucks*), 523 (*facts about the five main clouds*) and 524 (*how to erase scar?*).
- The queries in which the semantic retrieval did not outperform the keyword baseline seem to be those where the semantic information obtained by the query processing module was scarce. One such query would be 467 (*Show me all information about dachshund dog breeders*). **However, keyword baseline only rarely provides significantly better results than semantic search**. The effect of the semantic information coverage is studied in more detail in section 6.3.3.3.
- As we pointed out before, we have not evaluated the effect of complex queries (in terms of relationships) because TREC Web search evaluation topics are written for keyword-based search engines and do not consider this type of query expressivity. Future work should explore other IR standard evaluation benchmarks such as those ones used in the QA track, to evaluate the effect of complex queries in the performance of the different search engines. We hypothesize that, under this conditions, **the performance of the semantic retrieval would improve significantly relative to that of the others**.

6.3.3.1 Studying the impact of retrieved non-evaluated documents

Given a TREC topic and a document, there is one of three possibilities:

- The document is judged as a relevant result.
- The document is judged as an irrelevant result.
- The document has not been judged in the TREC collection. If semantic search retrieves it, our metrics treat it as irrelevant.

As Table 6.7 shows, **only 44% of the results returned by semantic retrieval had been previously evaluated** in the TREC collection. The unjudged documents, 66%, are therefore considered irrelevant. However, some of these results may be relevant, and therefore the performance of semantic retrieval might be better than reported.

Topic	Evaluated
451	44.6%
452	31.3%
454	49.4%
457	54.6%
465	38.5%
467	38.0%
476	50.6%
484	13.4%
489	51.6%
491	47.2%
494	57.3%
504	32.8%
508	62.8%
511	61.3%
512	39.8%
513	54.5%
516	47.5%
523	20.3%
524	47.6%
526	44.6%
Mean	44.4%

Table 6.7 Documents retrieved by semantic retrieval that are evaluated

Fig 6.10 shows the probability of a result returned by the semantic retrieval approach to be evaluated as function of its position. Results in the first positions have a very high probability. In other words, the first results returned by the semantic retrieval approach are very likely to have also been returned by at least one of the TREC search engines. This explains why unevaluated results are a significant issue for MAP but not for P@10.

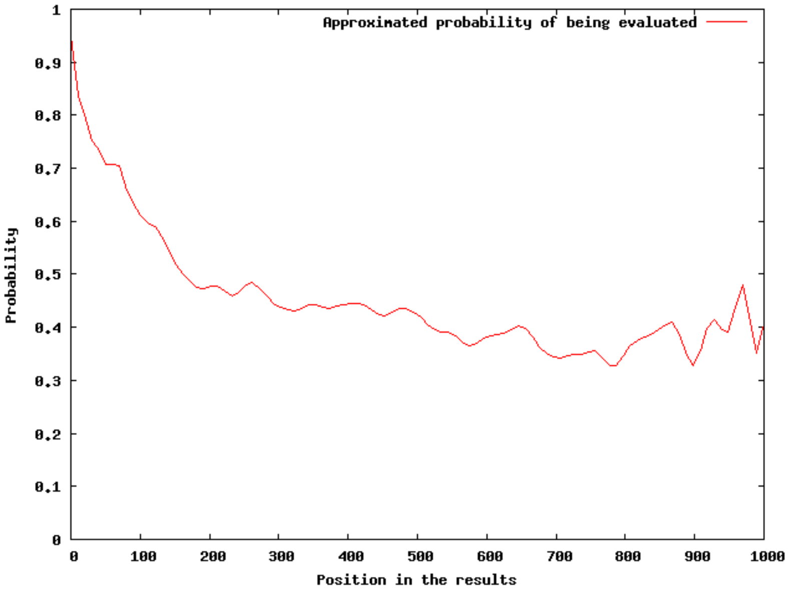


Fig 6.10 Probability of a document being evaluated by position

We now focus on how does the lack of evaluations for documents retrieved by semantic search affect the results by the MAP metrics? A legitimate question is whether the unevaluated results are actually relevant. Indeed, a result is unevaluated if it was not returned by any of the search engines in TREC, which one may expect to imply that it has a low probability of being relevant.

To provide a partial answer to this question we perform an informal evaluation of the first 10 unevaluated results returned for every query, a total number of 200 documents. 89% of these results occur in the first 100 positions for their respective query. We picked the first 10 because these are the most likely to be seen by the user and also because, occurring first on the query, they have a larger impact on the MAP measurements.

The results of our evaluation are shown in Table 6.8. For each query, we show the position in which the 10 documents we evaluated occurred. The positions with a result judged as relevant are shown in bold. We also show the percentage of these results that we judged as relevant and some notes that we gathered as we performed the evaluation.

Topic	Positions of top 10 unevaluated (by TREC) results										Relevance	Notes
451	25	26	27	28	32	34	35	36	37	38	0%	The documents are about cats but not about Bengal cats.
452	2	4	5	6	7	9	10	18	20	21	0%	It is not clear whether these documents are totally irrelevant since they talk about specific beaver's habitats, but in the context of salmon. However, this makes sense from a semantic retrieval engine perspective because the beaver dams are a nursery for salmon and therefore both habitats are strongly related.
454	**9**	**15**	22	**38**	**42**	**43**	**49**	**56**	**61**	**63**	90%	Documents are about specific symptoms and treatments for Parkinson and other degenerative disorders.
457	1	3	26	27	28	29	31	40	41	42	0%	Documents are about Chevrolet car models instead of Chevrolet trucks models.
465	4	5	8	**10**	**16**	**21**	25	**26**	27	**28**	50%	Documents are about specific diseases.
467	5	**6**	**7**	12	13	**14**	16	**17**	**20**	28	50%	All documents are about dog breeders but only some of them specifically talk about dachshunds.
476	**2**	3	7	**11**	12	**15**	**21**	**23**	24	25	50%	The relevant documents are about specific programs and movies related to Jennifer Anniston.
484	78	79	84	85	88	89	91	93	94	95	0%	Documents not related with Skoda.
489	**11**	54	68	79	**80**	82	**83**	97	105	106	30%	Some documents about calcium are from vendors and not from medical sources.

491	1	2	10	11	15	17	19	21	23	24	0%	Documents are about places where there have been tsunamis, but not specifically about them.
494	**86**	**88**	**128**	130	138	139	**140**	147	154	163	40%	Non relevant documents consider personal opinions of nirvana and related groups.
504	**2**	**4**	**5**	**6**	7	**8**	9	11	**12**	14	60%	Documents are about different types of algae.
508	4	**21**	**22**	23	**29**	**32**	39	41	48	**52**	50%	Documents are about specific diseases involved in hair loss.
511	**4**	27	**32**	**40**	**42**	**47**	48	**52**	60	**61**	70%	Documents are about specific diseases related with smoking.
512	23	**28**	30	31	33	35	**63**	65	66	**75**	30%	Some documents are about tornado effects but not about causes or atmospheric conditions.
513	61	62	**76**	108	129	**132**	**143**	150	153	**157**	40%	Relevant documents mainly about earthquakes and places where they usually occur.
516	46	71	72	76	77	87	88	**91**	96	100	10%	Documents provide pics and information about specific Halloween celebrations but not about the Halloween tradition.
523	14	21	22	**27**	**28**	29	37	**41**	43	45	30%	Documents are about microcrystal but not in the context of cloud generation.
524	**0**	13	14	**18**	19	**21**	50	59	60	61	30%	Most documents are about plastic surgery but not in the context of removing scar tissue.
526	1	11	32	72	79	98	100	101	107	108	0%	Some documents are about obesity but not about BMI.
Average:											31,5%	

Table 6.8 Results of top-10 retrieved unjudge documents evaluation

A significant portion, 31.5%, of the documents we judged turned out to be relevant. Clearly, this can not be generalized to all the unevaluated results returned by the semantic retrieval approach: as one moves towards the bottom, the probability of a result being relevant decreases, as shown by Fig 6.11. This figure is based only in the TREC evaluations, treating unevaluated (by TREC) results as irrelevant, so the actual probability is slightly higher. The figure shows that the probability of being relevant drops around the first 100 results and then varies very little. Regardless, we believe that the lack of evaluations for all the results returned by the semantic retrieval impairs its MAP value.

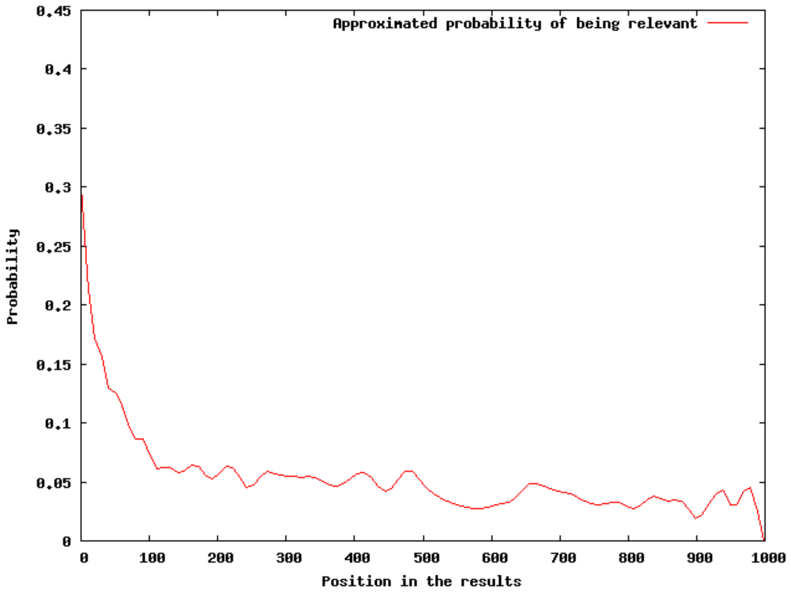


Fig 6.11 Probability of a document being relevant by position

The queries for which, of the top-10 documents retrieved that are not evaluated by TREC, we consider at least 50% relevant show that, in most cases, the semantic retrieval is obtaining new relevant documents when the query involves a class-instance relationship in the ontologies such as specific symptoms and treatments of Parkinson disease, specific movies or TV programs where Jenifer Anniston appears, etc. This effect was already highlighted in the previous chapter: semantic retrieval obtains better recall when querying for class instances.

Most of the results Table 6.8 lists, even those we consider irrelevant have related semantic information. For example, for topic 451, although documents about Bengal cats were not retrieved, most of the results were about other types of cats. For topic 457, the results centered around specifications of Chevrolet cars instead of Chevrolet trucks. This "potential recommendation" characteristic of our engine could even have a positive impact on the user's satisfaction, but this should be studied more carefully before definitive conclusions can be drawn.

6.3.3.2 Studying the trade-offs between the quality and quantity of annotations

As Table 6.5, Table 6.6 and Table 6.7 show, many relevant documents retrieved by the TREC search engines are not being retrieved by the semantic retrieval approach.

We hypothesize that the restrictions in the annotation process used may play a part here. Note that annotations are only generated when the contextual meaning of the entities in the ontology is found within the documents (see section 6.2.1.2). This loss of potential correct annotations is a price to be paid for the increase in accuracy.

We decided run a small-scale test with a variation of the annotation process based on NLP methodologies, as described in section 6.2.1.1. Although this new annotation method is less restrictive, given the fact that it relaxes the conditions to generate a new annotation, and that its weighting algorithm generates more accurate weights (it is based on an adaptation of TF-IDF over semantic entity frequencies), it is important to highlight that this annotation model is less feasible in terms of scalability than the one reported in section 6.2.1.2. Even though the annotation is an off-line process, the annotation model reported in section 6.2.1.2 is based on traditional keyword-based document indices, and therefore it can potentially take advantage of the structures used by large commercial search engines such as Google [53]or Yahoo[54].

With the aim to analyze the effect of this new annotation process, we randomly selected four topics, 452, 465, 476 and 484.

The results of the test are shown in Table 6.9 and Table 6.10. For each query, the tables include:

- The old value of the metric for the semantic retrieval approach.
- The new value of the metric for the semantic retrieval approach.
- The value of the metric for the Best TREC approach.
- The comparative between the new and the old value for the semantic retrieval approach.
- The comparative between the new semantic retrieval value and the Best TREC automatic value (when the Best TREC automatic value is not 0.0).

In the comparison (last two columns), values greater than 1 indicate that the new approach outperformed the other.

Topic	Old	New	TREC	New / Old	New / TREC
452	0.0383	0.0400	0.1952	1.04	0.2
465	0.1322	0.3200	0.0021	2.42	152.38
476	0.1265	0.3000	0.4131	2.37	0.73
484	0.1916	0.2300	0.0461	1.20	4.99
Averages:	0.1222	0.2225	0.1641	1.76	39.5753

Table 6.9 Quality of results by MAP using an NLP-based annotation model

Topic	Old	New	TREC	New / Old	New / TREC
452	0.2	0.2	0.3	1.0	0.67
465	0.3	0.5	0.0	1.7	
476	0.5	0.5	0.1	1.0	5
484	0.2	0.2	0.0	1.0	
Averages:	0.30	0.35	0.10	1.17	1.42

Table 6.10 Quality of results by P@10 using an NLP-based annotation model

As we can see in the tables, the quality of results increases significantly with the new annotation model. On average, by the MAP metric, the new model performs 1.76 times better than before. What is more, the quality of the first results, measured by P@10, did not diminish: in fact, it went up (albeit marginally).

[53] http://www.google.com/

[54] http://www.yahoo.com/

Interestingly, we can also see that relaxing the annotation conditions we can reduce the number of retrieved non-evaluated documents and made the results returned by the semantic retrieval significantly more likely to have been evaluated in the TREC collections. This is portrayed by Table 6.11, which shows an average increase in the number of evaluated results from 33% to 67%.

Topic	Old	New
452	31.30%	91.00%
465	38.50%	68.90%
476	50.60%	71.60%
484	13.40%	36.50%
Averages:	33.45%	67.00%

Table 6.11 Rate of retrieved evaluated documents for the two studied annotation processes

Although a bigger experiment should be performed to extract meaningful conclusions, we may say that, relaxing the annotation process has two positive effects: a) reduce the number of non-evaluated results and b) improve the quality of results for those queries where meaningful annotations where missing.

6.3.3.3 Studying the effect of the semantic coverage

Analyzing the results provided by Table 6.4, we saw that the semantic retrieval approach was generally outperforming the keyword-based methods, especially by the P@10 metric, for those cases where the semantic data was widely covering the information required by the query topic. However, in the evaluation, all the ontologies were selected from the SW and therefore, the information requested by the queries was, for most cases, just partially covered by the ontologies and KBs.

In order to study the effect of semantic coverage we ran another small-scale experiment to measure the impact of manually constructing ontologies and populating them from Wikipedia (as opposed to reusing previously existing SW ontologies). For this test, we picked the six TREC-9 topics where the best TREC automatic engine had more difficulties to provide relevant results: 453, 456, 468, 477, 478 and 483. Note that for all of these topics, the best TREC automatic approach did not retrieve any relevant result in the first top-10 positions. We manually generated six additional ontologies and populate them using Wikipedia to cover the domain of knowledge of those topics.

For this test we also used the NLP annotation model described in section 6.2.1.1. The results are provided in tables Table 6.12 and Table 6.13. For each query we show the performance by both metrics of the Lucene, semantic retrieval and Best TREC automatic approaches. The results for the Best TREC manual engine, which still outperform all the other engines for almost all queries, have been omitted from this table.

Maximal values for each query are shown in bold. The average results are shown at the bottom.

Topic	Lucene	Semantic retrieval	TREC automatic
453	0.2100	**0.2200**	0.1102
456	0.0100	**0.0200**	0.0073
468	0.0200	**0.0700**	0.0188
477	**0.0100**	**0.0100**	0.0045
478	**0.1200**	**0.1200**	0.0018
483	0.2000	**0.2300**	0.0624
Average	0.0950	**0.1117**	0.0342

Table 6.12 Effect of semantic coverage in the quality of results by MAP

Topic	Lucene	Semantic retrieval	TREC automatic
453	**0.2**	**0.2**	0.0
456	**0.0**	**0.0**	0.0
468	**0.0**	**0.0**	0.0
477	**0.0**	**0.0**	0.0
478	**0.4**	**0.4**	0.0
483	**0.3**	**0.3**	0.0
Average	**0.2**	**0.2**	0.0

Table 6.13 Effect of semantic coverage in the quality results by P@10

It is interesting to see that, by the P@10 metric, the semantic engine and Lucene performed identically and both of them outperformed the Best TREC automatic engine in all queries. This basically means that, the bad results obtained at P@10 by the Best TREC automatic approach are not due to the keyword-based retrieval methodologies, because Lucerne also outperforms it, but due to internal features of the Best TREC automatic ranking algorithms.

The results by the MAP metric are more interesting: the semantic engine outperformed Lucene in 2/3 of the queries and produced equal results in the rest, and outperformed the Best TREC automatic engine in all queries, a significant improvement over the results from the first experiment.

Although a bigger experiment should be performed to extract meaningful conclusions, following this small-scale experiment we may say that, as hypothesize, a good coverage of the semantic information can help to increase the overall performance of semantic retrieval algorithms.

6.4 Discussion

As was previously discussed in section 5.6, the vision of introducing ontologies as key enablers for semantically enhancing search engines on a decentralized, heterogeneous, dynamic and massive repository of content such as the Web is still an open problem. Three major limitations were identified towards the application of semantic retrieval models on the Web: usability, scalability and heterogeneity.

This chapter has shown several extensions that have been made to our original ontology-based retrieval model (explained in chapter 5) with the aim to take a step towards addressing above limitations, and test the feasibility of the semantic retrieval model in a large-scale heterogeneous environment.

To face the **usability** problem we have integrated PowerAqua (Lopez, Motta, & Uren, 2006) as the query-processing module of our system. PowerAqua is an ontology-based QA system capable of translating a NL query into ontological information, and extracting exact answers to user requests if enough semantic information is available. PowerAqua's ability to answer NL queries makes the user interface of our system more attractive than those of semantic search systems that rely on more complex ways of specifying information needs (e.g., through SPARQL queries). Thus, the integration of this tool into our system has brought two clear advantages. On the one hand, it copes with the usability limitation, allowing users to express their requirements using natural language, which, at the same time, provides a high level of expressivity with the queries. On the other hand, it can retrieve a concrete answer for the user when the appropriate semantic data is available.

The **scalability** problem has two faces. Firstly, it is necessary to scale the search algorithms to the large amount of unstructured content existing nowadays in the Web. Secondly, it is also desirable to exploit the increasing amount of available semantic metadata. We have proposed two automatic annotation mechanisms that semantically index the unstructured Web contents with semantic information.

- The first mechanism is based on NLP. It is less restrictive than the second approach, tending to create a bigger amount of annotations. It also implements a more accurate weighting annotation algorithm based on an adaptation of the TF-IDF measure. However, this annotation mechanism is computationally heavier, and therefore less scalable to a Web environment.
- The second mechanism is based on traditional keyword-based document indices, and therefore, it is more scalable since it could make use of the same indexing structures exploited by commercial search engines. Its annotation weighting mechanism provides less accurate results but, on the other hand, it is more flexible to changes in the ontologies and KBs. To increase the accuracy of the annotations, this mechanism exploits the semantic context of the entities. However, as shown in the experiments, this causes an important loss of potential correct annotations, which has a negative effect in the semantic retrieval algorithm.

The problem of **heterogeneity** is the most difficult to address. Web documents may contain information about any topic, and therefore, it would be desirable to have semantic knowledge covering any possible domain. However, although in the recent years, the SW area has contributed towards the development of more and better quality semantic information, the semantic information available nowadays is far from "complete".

To exploit this increasingly amount of semantic knowledge, we have integrated a SW gateway (explained in detail in chapter 7) into our semantic retrieval model. This module is responsible for collecting the semantic information available in the Web, and providing mechanisms for accessing and selecting the best semantic metadata according to users and applications needs. To alleviate the problem of heterogeneity, the query processing and annotation modules of our semantic retrieval system make use of this gateway.

The evaluation of this Web-extended semantic retrieval model aims to be a contribution on its own as well. Firstly, we have generated a widely applicable Web-scale evaluation benchmark for ontology-based retrieval models. This benchmark is the result of the adaptation of standard IR evaluation methodologies and datasets, and enhanced them with a significant degree of formality. Secondly, the experiments conducted take a step towards the advancements of applying ontology-based retrieval systems in large-scale and heterogeneous environments. The initial results of the comparative evaluation are promising, showing that when enough semantic information is available, the precision and average performance of the proposed semantic retrieval techniques improve, and only are worse than keyword-based search in very rare cases.

Several issues remain nonetheless open. One of the distinctive features of our system is its heterogeneity management. Indeed, unlike current systems, which are generally limited to a small set of domains by relying on a few pre-selected ontologies, our system can potentially cover a large amount of domains by making use of the ontologies available in the SW. Our empirical evaluation has shown however that the potential of our system is considerably constrained by the sparseness of the SW knowledge. In fact, we found that only 20% of the TREC topics include some concepts and predicates covered by online ontologies. Furthermore, most of the relevant ontologies were only weakly populated with instance data. While this status of the SW caused a suboptimal behavior of our system, any extension of the critical mass of ontologies and online available semantic data will result in a direct performance improvement of the proposed approach.

As a main conclusion of this chapter we point out the construction of a complete semantic retrieval approach that cover the entire IR process, from a NL query to a ranked set of documents. PowerAqua's ability to answer NL queries makes the user interface of our system more attractive than that of several semantic search prototypes which rely on more complex ways to specify an information need (e.g., SPARQL queries). Also, this system can retrieve a concrete answer when the appropriate semantic data is available. The semantic indexing and ranking modules of our system complement the query processing module in two ways. First, it provides a list of semantically ranked documents in addition to the concrete answer retrieved. Second, if the query processing module does not find any answer, the ranking module ensures that the system degrades gracefully to behave as a traditional keyword-based retrieval approach. At the time of writing we are not aware of any system that provides these functionalities.

Part III

Coping with semantic heterogeneity and incompleteness

Summary

The introduction of semantic knowledge to enhance IR models over heterogeneous information sources is not without its own tradeoffs and limitations. Web documents may contain information about any topic. Fine-grained semantic resources are fairly expensive to build, and formalization problems arise as semantic representations gain depth. The domain breath and depth coverage is hence bound to be far from complete. In order to cope with this inherent limitation, additional research has been undertaken in this thesis in a twofold direction. Dealing with semantic heterogeneity, Chapter 7 describes the definition of a SW gateway to external semantic resources on the Web , which has been integrated into our semantic retrieval model, providing access methods to semantic metadata, and selection strategies based on user input and application needs. Chapter 8 deals with knowledge incompleteness, which is tackled by a hybridation strategy, in which the results obtained from pure ontology-based retrieval algorithms, are blended with the output from a keyword-based retrieval model. Our research in this area focuses on the optimization of rank score normalization prior to a linear combination, based on statistical information characterizing the behaviour of the respective retrieval functions.

Chapter 7

Semantic knowledge gateway

This chapter focuses on the work carried out towards the generation of a SW gateway that collects, analyzes and gives access to available online semantic content, enabling the experimentation with the proposed retrieval algorithms on large amounts of semantic content. The rest of the chapter is organized as follows: section 7.1 provides a brief motivation through the construction of SW gateways. Section 7.2 explains the implemented structures to store and access the semantic content. Section 7.3 explains the developed algorithms for ontology evaluation and selection. The details of how this SW gateway is used in our semantic retrieval system are presented in section 7.4. To conclude, a brief discussion is presented in section 7.5.

7.1 Motivation

As reported in (D'Aquin, Gridinoc, Sabou, Angeletou, & Motta, 2007) the amount of published semantic knowledge in the SW, i.e., the number of available online ontologies and semantic documents is undergoing a steady growth. In (Motta & Sabou, 2006) the authors examine the future of SW applications and conclude that, in order to generate a successful new generation of semantic applications, the latter should be designed to exploit the growing body of currently available semantic markup. Leaving thus outside the applications the burden of creating the required semantic metadata, the focus is thus centred on finding and meaningfully combining the available semantic markup. Among the requirements that would characterize this new generation of systems we may highlight:

- **Semantic Data reuse vs. generation:** applications should be designed to operate with the semantic data that already exists. In other words, they should worry about providing mechanisms to exploit available semantic markup.
- **Multi-ontology vs. single-ontology systems:** applications should consume any number of ontologies and KBs at the same time. These systems assume that they operate on a large-scale SW characterized by huge amounts of heterogeneous data, which could be defined in terms of many different ontologies.
- **Scale as important as data quality:** applications should consider as key feature the size of the SW and be able to operate at large scale. Two important implications arise from this

emphasis on scale. Firstly the moment a system has to reason with very large amounts of heterogeneous semantic data, drawn from different sources, then necessarily these systems have to be prepared to accept variable data quality. Secondly, intelligence in these large-scale semantic systems becomes a side-effect of a system's ability to operate with large amounts of data, rather than being primarily defined by their reasoning ability.

- **Openness with respect to Web (non-semantic) resources:** a system that operates on a large-scale, rapidly evolving the SW, should also take into account the high degree of change of the conventional Web. E.g., annotation systems should work on any Web page, although of course, the quality of the annotation may degrade if there is not enough semantic metadata to cover the conceptual meanings of the Web page.

Giving a step towards the achievement of this new generation of semantic applications, it is desirable the generation of SW gateways that collect, analyze and give fast access to the online available semantic content. A SW gateway should accomplish three main goals:

- Collect the available semantic content from the Web.
- Implement efficient storage facilities to access the data.
- Implement ontology evaluation and selection algorithms to retrieve the most appropriate semantic information considering the user or application needs.

One of the most popular SW gateways currently available in the state of the art is **Swoogle** (Ding, Finin, Joshi, Pan, & Cost, 2004). This system claims to have indexed around ten thousand ontologies, which is a significant coverage of the SW data. However, the selection algorithms that this tool provides to users and applications are based on traditional IR methodologies, like the well known page-rank algorithm[55]. Thus, the ontology selection algorithms do not take into account semantic data quality measures such as lexical vocabulary, relations, consistency, correctness, etc.

Another very popular SW Gateway is **Watson** (D'Aquin, Baldassarre, Gridinoc, Angeletou, Sabou, & Motta, 2007). It combines the capabilities of Swoogle to crawl and search SW data with novel techniques to analyze the quality of content. This tool was under construction at the time of carrying out our experiments and it is currently being integrated into our system as a future work extension.

For the experiments reported in this thesis we generated our own SW gateway **WebCORE** (Fernández, Cantador, & Castells, 2006) (Cantador, Fernández, & Castells, 2007), but focusing our attention in the last two requirements. The collection of semantic content has been done manually from several ontology repositories and contains around 2GB of semantic metadata. The designed structures to store and access the semantic content, as well as the implemented algorithms for its evaluation and selection are explained in the following sections.

[55] http://es.wikipedia.org/wiki/PageRank

7.2 Storing and accessing semantic content

The proposed SW gateway, WebCORE (Fernández, Cantador, & Castells, 2006) (Cantador, Fernández, & Castells, 2007), focuses on providing users and applications with two main features: fast and efficient access to large amounts of semantic content and ability to manipulate several ontologies at the same time.

Following the first requirement, WebCORE is designed to pre-processes the gathered SW information and store it in several inverted indices using Lucene. These indexing structures allow applications to access basic lexical and taxonomical information quickly and efficiently.

Following the second requirement, WebCORE allows applications to manipulate multiple ontologies simultaneously under a common API. It also contains a cache structure to store the subset of ontologies that should be managed at a time.

7.2.1 Ontology indexing module

To efficiently access large amounts of SW content, WebCORE pre-processes and stores the gathered information in several inverted indices. Two kinds of indices are created, the **lexical ontology index** that associates each semantic entity (class, property, instance or literal) with a set of terms or lexical representations and, the **taxonomical ontology index** that associates each semantic entity with its direct subclasses and superclasses.

The lexical ontology-index generation is achieved by a concept-keyword extraction mechanism over the semantic entities. The keywords associated to each concept are extracted from the entity localName (which is part of its URI), the standard ontology meta property rdfs:label and optionally, from any other ontology property.

An example of the generated inverted index can be shown in Table 7.1 where each keyword is associated to one or several semantic entities from different ontologies. The semantic entities are uniquely identified within the system considering: the identifier of the ontology they belong to, their URI, their type (class, property, individual or literal) and their set of associated terms obtained after the concept-keyword extraction phase.

These indices are useful to identify, in a first step, the set of potential semantic entities (over the whole gathered SW content) that can be associated to a set of pre-defined terms describing a user query, a document, or any other application need.

keyword	Ontology Entities
Lorca	E2, E11, E120, …
Writer	E57, E62, E34, ..
Animal	E43, …

Table 7.1 Lexical ontology index.

To search the set of semantic entities associated to a specific term in the indices, we make use of the search capabilities of Lucene and the term relations obtained with WordNet (Fellbaum, 1998).

Lucene allows performing three different kinds of searches within the lexical ontology index:

- *Exact search*: the index must contain the exact searched term to retrieve an answer.
- *Fuzzy search*: the keywords stored in the index must be "similar" to the searched term. The similarity is computed using the Levenshtein distance (Levenshtein, 1966) and considering a established prefix that represents the number of letters that must be equal at the beginning of both words.
- *Spell search*: the searched term might contain some spelling mistakes. In this case, Lucene provides some suggestions of additional terms. For these cases, the system uses the first suggestion in order to perform a new search within the index.

WordNet allows extending the searched terms with three main types of relationships: synonyms, hypernyms and hyponyms. Searching for related terms increases the chances of finding a match within the index.

A second index level is generated to store taxonomical information. In this way, the ontology entities are also associated with its main superclasses and subclasses. An example of this indexing structure can be shown in Table 7.2.

Ontology Entity	Direct Subclasses	Direct Superclasses
E1	E11, E120	E3, E14, E22
E2	-	E3, E23 E41
E3	E2, E1, …	-

Table 7.2 Taxonomical ontology index

The lexical and taxonomical indices increase the mapping speed of semantically sound entities, allowing the management in real time of the distributed semantic information. For those cases in which the system requires more information than the one stored in the indices, the SW gateway provides a multi-ontology accessing module that allows managing several ontologies at a time within the application.

7.2.2 Multi-ontology access module

Providing universal access to multiple ontologies from different applications presents two main difficulties: accessing the semantic content in a common way for all the applications and generating appropriate multi-ontology management modules to administer several ontologies at a time.

Three main problems should be addressed in order to access the semantic content in a common way for all the applications:

- Ontologies are expressed in different query languages (RDF, OWL, DAML, etc)
- Ontologies can be stored in different types of repositories (databases, text files, URLs, etc)

- Different ontology frameworks implement different APIs to access ontologies (Sesame [56], Jena[57], etc)

To address this first problem we have developed a common API to access all the distributed semantic content. Figure 7.1 shows the architectural design and the set of layers involved in the semantic content accessing process.

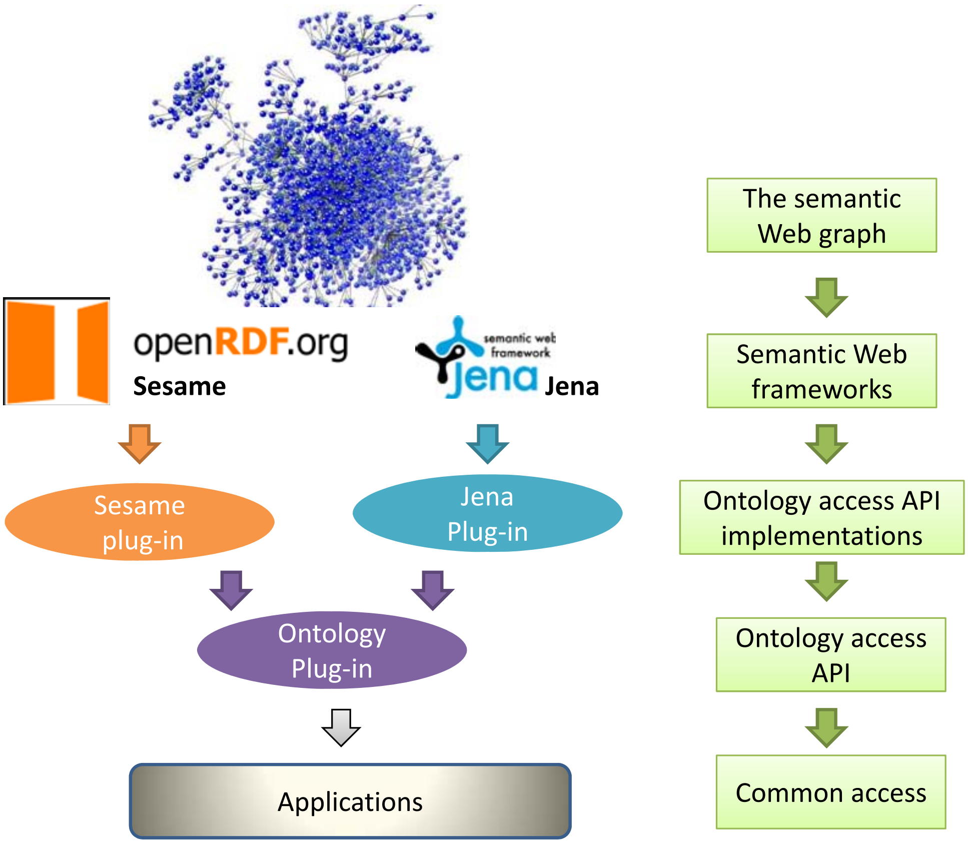


Fig 7.1 Common access to Semantic Web content[58]

In the first level we have the OntologyPlugin API. This API contains a common set of functionalities to query ontologies and KBs independently of their language, type of storage and location. In a second layer we have the implementations of this API using the most popular SW frameworks, in this case Sesame and Jena. Different extensions of the implementations are done for these frameworks to encapsulate the different ontology languages and types of storage. These implementations are done using the APIs and the query languages available for the different SW frameworks that are the ones directly accessing the SW graph of information.

[56] http://www.openrdf.org/

[57] http://jena.sourceforge.net/

[58] Ontology image extracted from: http://accuracyandaesthetics.com/wp-content/uploads/2006/12/ontology.jpg

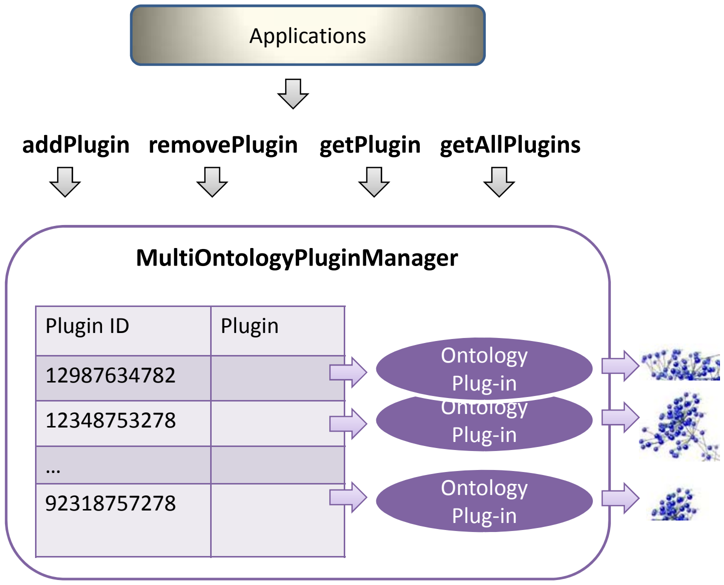
Loading a new ontology is as simple as create a new OntologyPlugin. To generate this structure the SW gateway needs the following information: the ontology identifier, its language, it corresponding framework and its location. An example of the information needed is shown in Fig 7.2. The SW gateway contains this information for the ontologies that have been previously gathered from the SW. However, any external application can provide information of a new ontology, so that the SW gateway can analyze and store it or access it at run time.

```
<ONTOLOGY>
   <ONTOLOGY_NAME>imdb-ontology.n3</ONTOLOGY_NAME>
   <ONTOLOGY_LANGUAGE>RDF</ONTOLOGY_LANGUAGE >
   <ONTOLOGY_PLUGIN_TYPE>JenaDB</ONTOLOGY_PLUGIN_TYPE>
   <ONTOLOGY_REPOSITORY>
          <REPOSITORY_SERVER_URL>dibus.ii.uam.es/ontologies/</REPOSITORY_SERVER_URL>
          <REPOSITORY_SERVER_PROXY/>
          <REPOSITORY_SERVER_PORT>3306</REPOSITORY_SERVER_PORT>
          <REPOSITORY_SERVER_LOGIN>atenea</REPOSITORY_SERVER_LOGIN/>
          <REPOSITORY_SERVER_PASSWORD>diosasabiduria</REPOSITORY_SERVER_ PASSWORD />
   </ONTOLOGY_REPOSITORY>
 </ONTOLOGY>
```

Fig 7.2 Information needed to create a new OntologyPlugin.

To manage several ontologies at a time in the application, the SW gateway provides an additional API to encapsulate a cache of ontologyPlugings. This API, named as MultiOntologyPlugin contain four basic functionalities: addPlugin (*OntologyPlugin op*), getPlugin(*OntologyPlugingIdentifier op*), removePluging(*OntologyPlugingIdentifier opi*) and *List<OntologyPlugin>* getAllPlugins()

Internally this cache structure is managed as a Hash table of OntologyPlugings, where each ontologyPluging is associated to the ontology identifier. A graphical view of the MultiOntologyPlugin structure can be shown in Fig 7.3.



Plugin ID	Plugin
12987634782	
12348753278	
…	
92318757278	

Fig 7.3 Architectural design of the MultiOntologyPlugin module

As we have shown, our SW gateway provides large-scale storage and accessing capabilities. It provides the necessary structures to manage multiple ontologies and KBs at the same time and common API to access the semantic content independently of the ontology-language, storage type and location.

On the top of this storage and accessing structures the SW gateway provides ontology evaluation measures that allow users and applications to discriminate or select the most appropriate available semantic content. The following section explains the design of our SW gateway ontology-evaluation platform.

7.3 Evaluating and selecting semantic content

Resources from open decentralized environments like the Web involve a naturally higher degree of imperfection, noise, and variable reliability, compared to more controlled settings. In such conditions, assessing the quality and suitability of semantic data, and selecting the most appropriate resources among the available ones, are a relevant issue. Once the available semantic information is collected, pre-processed and stored, ontology evaluation and selection algorithms must be implemented in order to retrieve the most appropriate semantic information for users and applications.

We have developed a set of collaborative ontology reuse and evaluation measures on top of the previously described ontology storage structures. These measures compute the similarity between a certain problem, or Golden Standard, and the set of available semantic data. The Golden Standard aims to simulate the behavior of a user or an application requesting semantic information. The system retrieves those ontologies that better match the Golden Standard according to content similarities and quality measures.

Content similarities are obtained by studying the lexical and the taxonomical structure of ontologies. Quality measures are introduced by taking advantage of collaborative evaluation strategies. In this way, we make use of the so-called "wisdom of the crowed" to obtain the best rated ontologies by prior ontology users, according to a selected set of quality criteria.

The following sections explain in detail the ontology-evaluation platform and measures.

7.3.1 Ontology evaluation platform

In this section we describe our ontology evaluation platform. Fig 7.4 shows an overview of the architecture. We distinguish three different modules. The first one, shown on the left side, receives the Golden Standard definition as a set of initial terms and allows the user to modify and extend it using WordNet (Miller G. , WordNet: A lexical database, 1995). The second one, in the center of the figure, allows the user to select a set of content ontology evaluation techniques provided by the system to recover the ontologies closest to the given Golden Standard. The third one, on the right, is a collaborative module that re-ranks the list of recovered ontologies, taking into consideration previous feedback and evaluations from the users.

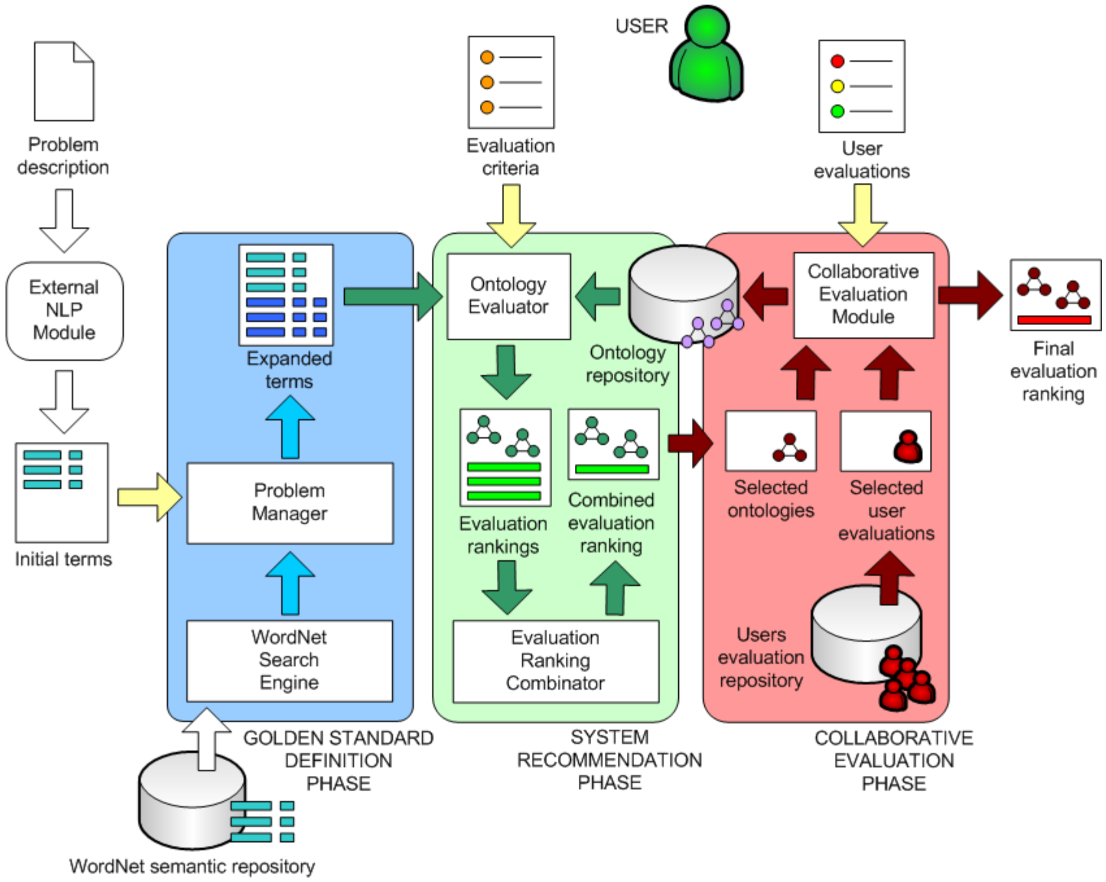
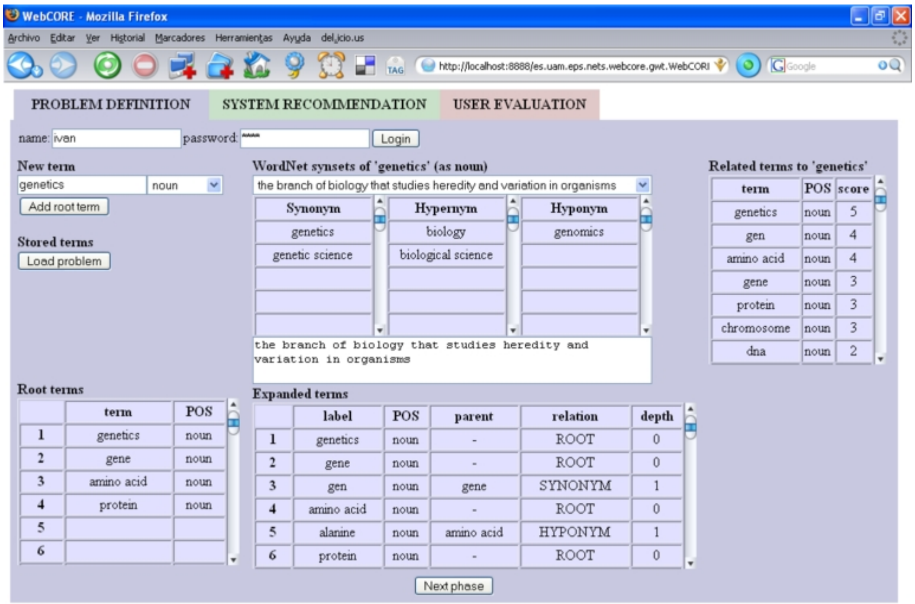

USER
Problem description
External NLP Module
Initial terms
Evaluation criteria
User evaluations
Expanded terms
Problem Manager
WordNet Search Engine
Ontology Evaluator
Ontology repository
Evaluation rankings
Combined evaluation ranking
Evaluation Ranking Combinator
Collaborative Evaluation Module
Final evaluation ranking
Selected ontologies
Selected user evaluations
Users evaluation repository
GOLDEN STANDARD DEFINITION PHASE
SYSTEM RECOMMENDATION PHASE
COLLABORATIVE EVALUATION PHASE
WordNet semantic repository

Fig 7.4 Ontology-evaluation platform architecture

7.3.1.1 Golden standard definition

The Golden Standard Definition module receives an initial set of terms. These set of terms might come from a Natural Language Query of an user, in which case, the system will retrieve the set of ontologies better covering the user needs. It might come from a Web document, in which case the system will retrieve the ontologies more suitable to perform an annotation process. It might come from a user profile, in which case the system will return the set of ontologies that better describe the user preferences, etc. To simulate a semi-automatic approach where terms may come from users and /or from automatic process the initial set of terms are obtained by an external Natural Language Processing module (Alfonseca, Moreno-Sandoval, Guirao, & Ruiz-Casado, 2006) from a set of documents related to the specific domain in which the user is interested. This NLP module would receive the repository of documents and return a list of pairs (lexical entry, part of speech), that roughly represents the domain of the problem. Once this initial set of automatic terms (or root terms) has been extracted, the system allows the user to expand the terms using WordNet (Miller G. , WordNet: A lexical database, 1995) and some of the relations it provides (hypernym, hyponym and synonym) in a semi-automatic way. The new terms added to the Golden Standard using these relations might also be extended again and added to the problem definition.


WebCORE - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.icio.us
http://localhost:8888/es.uam.eps.nets.webcore.gwt.WebCORE
Google
PROBLEM DEFINITION
SYSTEM RECOMMENDATION
USER EVALUATION
Login
Add root term
Load problem
Next phase

name: ivan password: ****

New term

genetics noun

Stored terms

WordNet synsets of 'genetics' (as noun)

the branch of biology that studies heredity and variation in organisms

Synonym	Hypernym	Hyponym
genetics	biology	genomics
genetic science	biological science	

```
the branch of biology that studies heredity and
variation in organisms
```

Related terms to 'genetics'

term	POS	score
genetics	noun	5
gen	noun	4
amino acid	noun	4
gene	noun	3
protein	noun	3
chromosome	noun	3
dna	noun	2

Root terms

	term	POS
1	genetics	noun
2	gene	noun
3	amino acid	noun
4	protein	noun
5		
6		

Expanded terms

	label	POS	parent	relation	depth
1	genetics	noun	-	ROOT	0
2	gene	noun	-	ROOT	0
3	gen	noun	gene	SYNONYM	1
4	amino acid	noun	-	ROOT	0
5	alanine	noun	amino acid	HYPONYM	1
6	protein	noun	-	ROOT	0

Fig 7.5 Golden Standard definition phase

The final representation of the Golden Standard can be defined as a set of terms T(L^G, POS, L^{GP}, R, Z) where:

- L^G is the set of lexical entries defined for the Golden Standard (root terms).
- POS corresponds to the different Parts Of Speech considered by WordNet: *noun*, *adjective*, *verb* and *adverb*.
- L^{GP} is the set of lexical entries of the Golden Standard that have been extended.
- R is the set of relations between terms of the Golden Standard: *synonym*, *hypernym*, *hyponym* and *root* (if a term has not been obtained by expansion, but is one of the initial terms).
- Z is an integer number that represents the depth or distance of a term to the root term from which it has been derived.

Example:

T_1 ("pizza", noun, "", ROOT, 0). T_1 is one of the root terms of the Golden Standard. The lexical entry that it represents is "pizza", its part of speech is "noun", it has not been expanded from any other term so its lexical parent is the empty string, its relation is ROOT and its depth is 0.

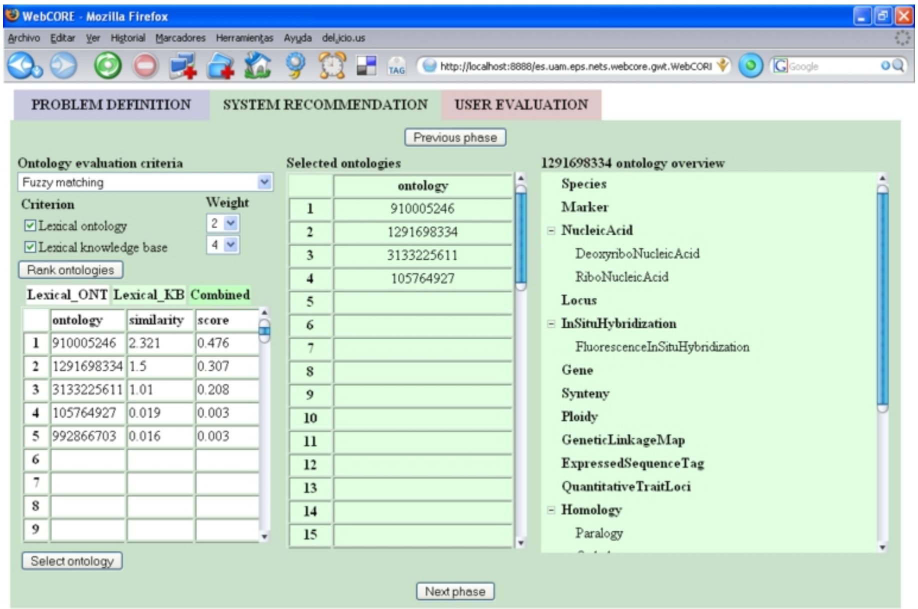
T_2 ("pizza pie", noun, "pizza", Synonym, 1). T_2 is a term expanded from T_1. The lexical entry it represents is "pizza pie", its part of speech is "noun", the lexical entry of its parent is "pizza", it has been expanded by the synonym relation and the number of relations that separated it from the root term T_1 is 1.

Fig 7.5 shows the interface of the Golden Standard Definition phase. In the left part we can see the list of root terms. The user is allowed to manually insert new root terms giving their lexical entries and selecting their parts of speech. The correctness of these new insertions is controlled by verifying that all the considered lexical entries belong to the WordNet (Miller G. , WordNet: A lexical database, 1995) repository. In the central bottom level we can see the final Golden Standard definition: the final list of (root and expanded) terms that represent the domain of the problem. In the central top level it can be seen how the user can make a term expansion. The user selects one of the previous terms from the Golden Standard definition and the system shows him all its meanings contained in WordNet (Miller G. , 1995). After he chooses one, the system automatically presents him three different lists with the synonyms, hyponyms and hypernyms of the term. The user can choose one or more elements of these lists and they will automatically be added to the expanded terms list. For each expansion the depth of the new term is increased by one unit. This will be used later to measure the importance of the term within the Golden Standard: the greater the depth of the derived term with respect to its root term, the less its relevance will be.

7.3.1.2 Content similarity evaluation

In this phase the system should retrieve the ontologies that better conceptualize the Golden Standard domain attending to lexical and taxonomical similarities. Fig 7.6 represents the structure of the content similarity evaluation phase of the system. Firstly, the user selects a set of content evaluation criteria to be performed. After considering the selected criteria and taking into account the Golden Standard and the previously gathered ontologies, the system retrieves a ranked list of ontologies (ordered by their similarity to the Golden Standard) for each criterion. Then, all these lists are merged using rank fusion techniques (chapter 8) to obtain a global measure.

Fig 7.6 represents the user interface of the content similarity evaluation module. In the left upper level we distinguish the criteria selection phase. By now, two content evaluation criteria can be selected to retrieve the most similar ontologies: 1) the **lexical criterion**, which measures similarity between the lexical entries of the Golden Standard and the lexical entries of the ontologies and, 2) the **taxonomic criterion**, which evaluates the hierarchical structure between them. The user can also select the relevance of each criterion in the rank aggregation process, using a range of discrete values [1, 2, 3, 4, 5], where 1 symbolizes the lowest relevance value and 5 the highest. These measures are explained in section 7.3.2 of this chapter. The left bottom level of Fig 7.6 shows a different ranked list for each criterion and the final fused list. In each of these tables, two different ratings are displayed for each ontology. The first one refers to the similarity between the ontology and the Golden Standard. The second rating, score, shows the similarity value normalized by the sum of all the values. The score measure exhibits the distribution of the ratings and allows us to better evaluate the different techniques. Once the final ranked list has been retrieved, the system allows the user to select a subset of ontologies that he considers adequate for the Collaborative Evaluation Phase. In the case of applications, ontologies are automatically selected from the ranking considering either a predefined number of ontologies, a score value higher than a predefined threshold or both criteria at the same time.


WebCORE - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.icio.us
http://localhost:8888/es.uam.eps.nets.webcore.gwt.WebCORI
Google
PROBLEM DEFINITION
SYSTEM RECOMMENDATION
USER EVALUATION
Previous phase
Rank ontologies
Select ontology
Next phase

Ontology evaluation criteria: Fuzzy matching

Criterion	Weight
☑ Lexical ontology	2
☑ Lexical knowledge base	4

Lexical_ONT Lexical_KB Combined

	ontology	similarity	score
1	910005246	2.321	0.476
2	1291698334	1.5	0.307
3	3133225611	1.01	0.208
4	105764927	0.019	0.003
5	992866703	0.016	0.003
6			
7			
8			
9			

Selected ontologies

	ontology
1	910005246
2	1291698334
3	3133225611
4	105764927
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

1291698334 ontology overview

- Species
- Marker
- NucleicAcid
  - DeoxyriboNucleicAcid
  - RiboNucleicAcid
- Locus
- InSituHybridization
  - FluorescenceInSituHybridization
- Gene
- Synteny
- Ploidy
- GeneticLinkageMap
- ExpressedSequenceTag
- QuantitativeTraitLoci
- Homology
  - Paralogy

Fig 7.6 Content similarity evaluation phase

7.3.1.3 Quality evaluation

This module has been designed to face the challenge of evaluating those ontology features that are by their nature, more difficult for machines to address. Where human judgment is required, the system will attempt to take advantage of Collaborative Filtering techniques (Masthoff, 2004) (Montaner, Lopez, & De la Rosa, 2004) (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). Some approaches for ontology development (Sure, Erdmann, Angele, Staab, Studer, & Wenke, 2002) have been presented in the literature concerning collaboration techniques. However to our knowledge, Collaborative Filtering strategies have not been used in the context of ontology evaluation and reuse.

Collaborative filtering strategies make automatic predictions (filter) about the interests of a user by collecting taste information from many users (collaborating).

In our evaluation platform, a new ontology evaluation measure based on collaborative filtering is proposed, considering user's interest and previous assessments of the ontologies. The collaborative module implemented in this work ranks and presents the best ontologies for the user, taking into consideration previous manual evaluations. Several issues have to be considered in a collaborative system. The first one is the representation of user profiles. The type of user profile selected for our system is a user-item rating matrix *(ontologies evaluations based on specific criteria)*. The initial profile is designed as a manual selection of five predefined criteria (Paslaru, 2005):

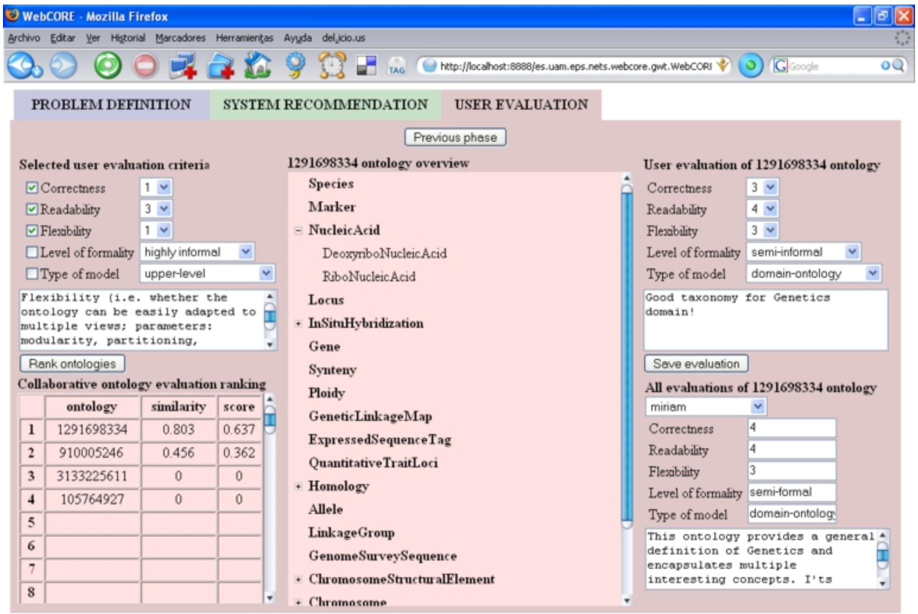
- Correctness: specifies whether the information stored in the ontology is true, independently of the domain of interest.
- Readability: indicates the non-ambiguous interpretation of the meaning of the concept names.
- Flexibility: points out the adaptability or capability of the ontology to change.
- Level of Formality: highly informal, semi-informal, semi-formal, rigorously-formal.

- Type of model: upper-level (for ontologies describing general, domain-independent concepts), core-ontologies (for ontologies describing the most important concepts on a specific domain), domain-ontologies (for ontologies describing some domain of the world), task-ontologies (for ontologies describing generic types of tasks or activities) and application-ontologies (for ontologies describing some domain in an application-dependent manner).

The above criteria can be divided in two different groups: 1) the *discrete* criteria (correctness, readability and flexibility) that are represented by discrete numeric values [0, 1, 2, 3, 4, 5] where 0 indicates that the ontology does not fulfill the criterion, and 5 indicates the ontology completely satisfies the criterion and, 2) the *boolean* criteria (level of formality and type of model) are represented by a specific value that is either satisfied by the ontologies, or not. The collaborative system does not implement any profile learning technique or relevance feedback to update user profiles. But, the profiles may be modified manually.

Ontologies (our content items) are recommended based on previous users' evaluations. To evaluate the levels of relevance of the ontologies, our User Profile-Item matching technique will make comparisons between the user's interests and the ontology's evaluations stored into the system. This will be explained in section 7.3.2.2.

Fig 7.7 shows the Collaborative Evaluation module. At the left top level the user's interest can be selected as a subset of criteria with associated values representing thresholds that manual evaluations of the ontologies should fulfil. For example, when a user sets a value of 3 for the correctness criterion, the system recognizes that he is looking for ontologies whose correctness value is greater than or equal to 3. Once the user's interests have been defined, the set of manual evaluations stored in the system is used to compute which ontologies fit his interest best. The left bottom level shows the final ranked list of ontologies returned by the Collaborative Filtering module. To add new evaluations to the system, the user must select an ontology from the list and choose one of the predetermined values for each of the five aforementioned criteria (right part of the figure). The system also allows the user to add some comments to the ontology evaluation in order to provide more feedback. On the right bottom side, we can also see how the system enables the user to observe all the evaluations stored into the system about a specific ontology. This may be of interest since we may trust some users more than others.


WebCORE - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda del.icio.us
http://localhost:8888/es.uam.eps.nets.webcore.gwt.WebCORI
Google
PROBLEM DEFINITION
SYSTEM RECOMMENDATION
USER EVALUATION
Previous phase
Rank ontologies
Save evaluation

Selected user evaluation criteria

☑ Correctness 1
☑ Readability 3
☑ Flexibility 1
□ Level of formality highly informal
□ Type of model upper-level

Flexibility (i.e. whether the ontology can be easily adapted to multiple views; parameters: modularity, partitioning,

Collaborative ontology evaluation ranking

	ontology	similarity	score
1	1291698334	0.803	0.637
2	910005246	0.456	0.362
3	3133225611	0	0
4	105764927	0	0
5			
6			
7			
8			

1291698334 ontology overview

Species
Marker
⊟ NucleicAcid
DeoxyriboNucleicAcid
RiboNucleicAcid
Locus
+ InSituHybridization
Gene
Synteny
Ploidy
GeneticLinkageMap
ExpressedSequenceTag
QuantitativeTraitLoci
+ Homology
Allele
LinkageGroup
GenomeSurveySequence
+ ChromosomeStructuralElement
+ Chromosome

User evaluation of 1291698334 ontology

Correctness 3
Readability 4
Flexibility 3
Level of formality semi-informal
Type of model domain-ontology

Good taxonomy for Genetics domain!

All evaluations of 1291698334 ontology

miriam

Correctness 4
Readability 4
Flexibility 3
Level of formality semi-formal
Type of model domain-ontolog

This ontology provides a general definition of Genetics and encapsulates multiple interesting concepts. I'ts

Fig 7.7 Quality evaluation phase

7.3.2 Evaluation measures

This section presents the two kinds of evaluation measures: a) the content-based similarity measures, that are automatically computed considering internal ontology features such as lexical and taxonomical information and b) the quality measures that are automatically computed using collaborative filtering techniques that consider previous user feedback.

7.3.2.1 Content-based measures

In order to obtain similarities between the Golden Standard and the stored ontologies based on the content of the latter, two different levels have been considered, the lexical and the taxonomic. Several measures have been developed and tested for each level. In the following sections we present the approaches that have shown better performance.

7.3.2.1.1 Lexical measures

The lexical evaluation assesses the similarity between the domain of the problem as described by the Golden Standard and an ontology by comparing the lexical entries, or words that represent them. A new lexical evaluation measure based on Maetche and Staab work (Maedche & Staab, 2002) will be explained in this section. Some definitions must first be introduced.

- **Definition 1** A ***lexical entry*** *l* represents a string or word.
- **Definition 2** The ***Golden Standard Lexicon*** L^G is defined by the set of lexical entries extracted from the terms of the Golden Standard, where each term has a single lexical entry that represents it.

- **Definition 3** An ***Ontology Lexicon*** L^O is defined as the set of lexical entries extracted from the Concepts of the Ontology. Each concept is represented by one or more lexical entries that are extracted from the concept name, the rdfs:label property value, or other properties that could be added to the lexical extraction process considering the characterization of each ontology.
- **Definition 4** The ***Levenshtein distance***, $ed(l_i, l_j)$ between two lexical entries l_i and l_j measures the minimum number of token insertions, deletions and substitutions to transform l_i into l using a dynamic algorithm. Example: ed ("pizzapie", "pizza_pie") = 1

Maedche and Staab (Maedche & Staab, 2002) propose a lexical similarity measure for strings called String Matching. This method compares two lexical entries l_i, taking into account the Levenshtein distance against the shortest lexical entry.

$$SM(l_i, l_j) = \max(0, \frac{\min(|l_i|, |l_j|) - ed(l_i, l_j)}{\min(|l_i|, |l_j|)}) \in [0,1]$$

SM returns a degree of similarity between 0 and 1, where 0 is a null match and 1 represents a perfect match. Example: SM("pizzapie", "pizza_pie") = 7/8.

Based on the String Matching they propose a lexical similarity measure to compare an ontology to a Golden Standard, by computing the average string matching between the set of Golden Standard lexical entries and the set of ontology lexical entries:

$$\overline{SM}(L^G, L^O) = \frac{1}{|L^G|} \sum_{l_i \in L^G} max_{l_j \in L^O} SM(l_i, l_j)$$

$\overline{SM}(L^G, L^O)$ is an asymmetric measure that determines the extent to which the lexical level of the Golden Standard is covered by the lexical level of the Ontology. Future work must be done in order to penalize those ontologies which contain all the strings of the Golden Standard but also many others. There is one principle difference between that approach and ours; Maedche defines the Golden Standard as an ontology, while we use our own model. This fact provides us with the capability to use all the additional information stored in the Golden Standard in order to improve content evaluation measures. When a domain is modelled as a set of lexical entries, some lexical entries have greater relevance when defining the semantics than do others. Assuming this characteristic we have decided to distinguish the importance of the Golden Standard terms. The root terms are considered the most representative ones while the relevance of the expanded terms depends on the number of relations that separate them from a root term. With this modification we emphasize the main semantics and relegate the complementary ones into the background. In this work we define the Golden Standard Lexical weight measure to evaluate the importance of each term.

- **Definition 5** Given a list of lexical entries $L = \{l_i\}$ expanded from a common root lexical entry, we define the ***Golden Standard lexical weight of*** $l \in L$ as:

$$w(l) = \begin{Bmatrix} 1 + \dfrac{max_i(Depth(l_i)) - Depth(l)}{max_i(Depth(l_i))} \in [1,2] & if|L| > 1 \\ 2 & otherwise \end{Bmatrix}$$

The value returned is represented as a degree of relevance between 1 (the farthest distance to the root lexical entry), and 2 (no distance to the root lexical entry). If the root lexical entry has not been expanded we assign it the weight value 2.

Fig 7.8 shows an example of this measure, where T_1 is the root term and consequently has the greater weight. T_3 is the most remote term and it has the smaller weight. The intermediate terms like T_2 have a weight between the maximum and the minimum relative to their distance from the root term.

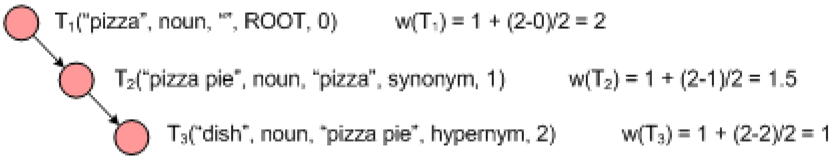

T₁("pizza", noun, "", ROOT, 0)
w(T₁) = 1 + (2-0)/2 = 2
T₂("pizza pie", noun, "pizza", synonym, 1)
w(T₂) = 1 + (2-1)/2 = 1.5
T₃("dish", noun, "pizza pie", hypernym, 2)
w(T₃) = 1 + (2-2)/2 = 1

Fig 7.8 Golden Standard lexical weight measure

In our approach, we have modified the previous lexical measure taking into account the weight or relevance of each term to represent the semantics of the domain.

$$\overline{SM}(L^G, L^O) = \frac{1}{|L^G|} \sum_{l_i \in L^G} max_{l_j \in L^O} SM(l_i, l_j) \cdot w(l_i)$$

Through our experiments, this new measure has been shown to better discriminate the ontologies, giving a higher similarity value to the ontologies that are closer to the Golden Standard and lower rating to the ontologies that worst fit the problem domain. Future work is needed in order to give more or less relevance to the derived terms of the Golden Standard using not only their distances to the root terms but also, the kind of relation from which they have been derived, synonym, hypernym or hyponym.

7.3.2.1.2 Taxonomic measures

The taxonomic evaluation assesses the degree of overlapping between the hierarchical structure of the ontology, defined by the "is-a" relation and the Golden Standard structure, defined by the derivations of terms to complete the domain representation. The following notations and definitions will be used to define our measure:

- $\boldsymbol{T_i^G} \in \boldsymbol{T^G}$represents a Golden Standard term.
- $\boldsymbol{C_i^O} \in \boldsymbol{C^O}$represents an Ontology concept.
- **Definition 8** The ***Semantic Cotopy of a Golden Standard term*** $SC(T_i^G)$ is defined as the set of lexical entries of the terms derived from the same root term asT_i^G, including the lexical entries of T_i^G.
- **Definition 9** The ***Semantic Cotopy of an Ontology concept*** $SC(C_i^O)$ is defined as the set of lexical entries of the concepts related with C_i^O in the ontology with a direct relation, including the lexical entries of C_i^O.

Given Maedche and Staab (Maedche & Staab, 2002) measure, an adaptation is performed to obtain a similarity between an ontology Concept and a Golden Standard Term relative to the new Golden Standard definition. The similarity is computed as the intersection between the Semantic Cotopy of the Golden Standard term and the Semantic Cotopy of the ontology concept normalized by the total possible overlap.

$$TS(T_i^G, C_i^O) = \frac{|SC(T_i^G) \cap SC(C_i^O)|}{|SC(T_i^G) \cup SC(C_i^O)|}$$

In order for two lexical entries to be considered a match, their similarity must be greater than a threshold empirically estimated as 0.2. For similarities below this value we have observed there is no significant morphological resemblance between terms.

The taxonomic similarity measure considers all the overlaps between the Ontology and the Golden Standard. In order to optimize the evaluation, only a subset of terms and concepts are used to assess the taxonomic similarity. This subset is obtained through the lexical measurement, this is done by selecting only the terms and concepts that have matched with a similarity value greater than 0.2.

$$\overline{TS}(T^G, C^O) = \frac{1}{|T^G|} \sum_{\substack{T_i^G \in T^G, C_j^O \in C^O \; and \\ \exists l_i \in T_i^G, \exists l_j \in C_j^O : SM(l_i, l_j) > 0{,}2}} TS(T_i^G, C_j^O)$$

7.3.2.2 Quality measures

In this section, we describe a novel ontology recommendation algorithm that exploits the advantages of collaborative filtering, and explores the manual evaluations stored in the system, for ranking the set of ontologies that best fulfils the user or application interests.

As we explained in section 7.3.1.3, user evaluations are represented as a set of five defined criteria and their respective values, manually determined by the users who made the evaluations. These criteria can have discrete numeric or non-numeric values. Moreover, user interests are expressed like a subset of the above criteria, and their respective values, meaning thresholds or restrictions to be satisfied by user evaluations. Thus, a numeric criterion will be satisfied if an evaluation value is equal or greater than that expressed by its interest threshold, while a non-numeric criterion will be satisfied only when the evaluation is exactly the given "threshold" (i.e. in a Boolean or yes/no manner).

According to both types of user evaluation and interest criteria, *numeric* and *Boolean*, the recommendation algorithm will measure the degree in which each restriction is satisfied by the evaluations, and will recommend a ranked ontology list according to similarity measures between the thresholds and the collaborative evaluations. To create the final ranked ontology list the recommender module follows two phases. In the first one it calculates the similarity degrees between all the user evaluations and the specified interest criteria thresholds (set by users or applications). In the second one it combines the similarity measures of the evaluations, generating the overall rankings of the ontologies.

As we explained before, in the system a user evaluate a specific ontology considering five different criteria (see section 7.3.1.3). These five criteria can be divided in two different groups: 1) the *numeric* criteria (correctness, readability and flexibility), which take discrete numeric values [0, 1, 2, 3, 4, 5], where 0 means the ontology does not fulfil the criterion, and 5 means the ontology completely satisfy the criterion, and, 2) the *Boolean* criteria (level of formality and type of model), which are represented by specific non-numeric values that can be or not satisfied by the ontology. Thus, user interests are defined as a subset of the above criteria, and their respective values representing the set of thresholds that should be reached by the ontologies.

Given a set of user or application interests, the system will size up all the stored evaluations, and will calculate their similarity measures. To explain these similarities we shall use a simple example of six different evaluations (E_1, E_2, E_3, E_4, E_5 and E_6) of a certain ontology. In the explanation we shall distinguish between the numeric and the Boolean criteria.

We start with the Boolean ones, assuming two different criteria, C_1 and C_2, with three possible values: “A”, “B” and “C”. In Table 1 we show the “threshold” values established by a user or application for these two criteria, “A” for C_1 and “B” for C_2, and the six evaluations stored in the system.

		Evaluations					
Criteria	**Thresholds**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**
C_1	**“A”**	“A”	“B”	“A”	“C”	“A”	“B”
C_2	**“B”**	“A”	“A”	“B”	“C”	“A”	“A”

Table 7.3 Threshold and evaluation values for Boolean criteria C_1 and C_2

In this case, because of the threshold of a criterion *n* is satisfied or not by a certain evaluation *m*, their corresponding similarity measure is simply 0 if they have the same value, and 2 otherwise.

$$similarity_{bool}(criterion_{mn}) = \begin{Bmatrix} 0 \text{ } if \text{ } evaluation_{mn} \neq \text{ } threshold_{mn} \\ 2 \text{ } if \text{ } evaluation_{mn} = \text{ } threshold_{mn} \end{Bmatrix}$$

The similarity results for the Boolean criteria of the example are shown in Table 7.4.

		Evaluations					
Criteria	**Thresholds**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**
C_1	**“A”**	**2**	**0**	**2**	**0**	**2**	**0**
C_2	**“B”**	**0**	**0**	**2**	**0**	**0**	**0**

Table 7.4 Similarity values for Boolean criteria C_1 and C_2

For the numeric criteria, the evaluations can overcome the thresholds to different degrees. Table 3 shows the thresholds established for criteria C_3, C_4 and C_5, and their six available evaluations. Note that E_1, E_2, E_3 and E_4 satisfy all the criteria, while E_5 and E_6 do not reach some of the corresponding thresholds.

		Evaluations					
Criteria	Thresholds	E1	E2	E3	E4	E5	E6
C_3	≥ 3	3	4	5	5	2	0
C_4	≥ 0	0	1	4	5	0	0
C_5	≥ 5	5	5	5	5	4	0

Table 7.5 Threshold and evaluation values for numeric criteria C_3, C_4 and C_5

In this case, the similarity measure has to take into account two different issues: the degree of satisfaction of the threshold, and the difficulty of achieving its value. Thus, the similarity between the value of criterion *n* in the evaluation *m*, and the threshold of interest is divided into two factors: 1) a similarity factor that considers whether the threshold is surpassed or not, and, 2) a penalty factor which penalizes those thresholds that are easier to be satisfied.

$$similarity_{num}(criterion_{mn}) = 1 + \ similarity^{*}_{num}(criterion_{mn}) \cdot penalty_{num}(threshold_{mn}) \in [0,2]$$

This measure will also return values between 0 and 2. The idea of returning a similarity value between 0 and 2 is inspired on other collaborative matching measures (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994) to not manage negative numbers, and facilitate, as we shall show in the next subsection, a coherent calculation of the final ontology rankings.

The similarity assessment is based on the distance between the value of the criterion *n* in the evaluation *m*, and the threshold indicated in the user's interests for that criterion. The more the value of the criterion *n* in evaluation *m* overcomes the threshold, the greater the similarity value shall be.

Specifically, following the expression below, if the difference *dif = (evaluation – threshold)* is equal or greater than 0, we assign a positive similarity in (0,1] that depends on the maximum difference *maxDif = (maxValue – threshold)* we can achieve with the given threshold; and else, if the difference *dif* is lower than 0, we give a negative similarity in [-1,0), punishing the distance of the value with the threshold.

$$similarity^{*}_{num} = (criterion_{mn}) = \left\{ \begin{matrix} \dfrac{1 + dif}{1 + maxDif} \in (0,1] \ \ if\ dif\ \geq 0 \\ \dfrac{dif}{threshold} \in [-1,0)\ \ if\ dif < 0 \end{matrix} \right\}$$

Table 7.6 summarizes the similarity* values for the three numeric criteria and the six evaluations.

		Evaluations					
Criteria	Thresholds	E1	E2	E3	E4	E5	E6
C_3	**≥ 3**	1/4	2/4	3/4	3/4	-1/3	-1
C_4	**≥ 0**	1/6	2/6	5/6	1	1/6	1/6
C_5	**≥ 5**	1	1	1	1	-1/5	-1

Table 7.6 Similarity* values for numeric criteria C_3, C_4 and C_5

Comparing the evaluation values of Table 7.5 with the similarity values of Table 7.6, the reader may notice several important facts:

- Evaluation E_4 satisfies criteria C_4 and C_5 with assessment values of 5. Applying the above expression, these criteria receive the same similarity of 1. However, criterion C_4 has a threshold of 0, and C_5 has a threshold equal to 5. As it is more difficult to satisfy the restriction imposed to C_5, this one should have a greater influence in the final ranking.
- Evaluation E_6 gives an evaluation of 0 to criteria C_3 and C_5, not satisfying either of them and generating the same similarity value of -1. Again, because of their different thresholds, we should distinguish their corresponding relevance degrees in the rankings.

For these reasons, a threshold penalty factor is applied, reflecting how difficult it is to overcome the given thresholds. The more difficult to surpass a threshold, the lower the penalty value shall be.

$$penality_{num}(threshold) = \frac{1 + threshold}{1 + maxValue} \in (0, 1]$$

Table 7.7 shows the threshold penalty values for the three numeric criteria and the six evaluations.

		Evaluations					
Criteria	**Thresholds**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**
C_3	**≥ 3**	4/6	4/6	4/6	4/6	4/6	4/6
C_4	**≥ 0**	1/6	1/6	1/6	1/6	1/6	1/6
C_5	**≥ 5**	1	1	1	1	1	1

Table 7.7 Threshold penalty values for numeric criteria C_3, C_4 and C_5

The similarity results for the numeric criteria of the example are shown in Table 7.8.

		Evaluations					
Criteria	**Thresholds**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**
C_3	**≥ 3**	**1.17**	**1.33**	**1.5**	**1.5**	**0.78**	**0.33**
C_4	**≥ 0**	**1.03**	**1.05**	**1.14**	**1.17**	**1.03**	**1.03**
C_5	**≥ 5**	**2**	**2**	**2**	**2**	**0.5**	**0**

Table 7.8 Similarity values for numeric criteria C_3, C_4 and C_5

As a preliminary approach, we calculate the similarity between an ontology evaluation and the application's or user's requirements as the average of its *N* criteria similarities.

$$similarity(evaluation_m) = \frac{1}{N} \sum_{n=1}^{N} similarity(criterion_{mn})$$

A weighted average could be even more appropriate, and might make the collaborative recommender module more sophisticated and adjustable to user needs. This will be considered for a possible enhancement of the system in the continuation of our research.

Once the similarities are calculated taking into account the user's or application's interests and the evaluations stored in the system, a ranking is assigned to the ontologies. The ranking of a specific ontology is measured as the average of its M evaluation similarities. Again, we do not consider different priorities in the evaluations of several users.

$$ranking(ontology) = \frac{1}{M}\sum_{m=1}^{M} similarity(evaluation_m) = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} similarity(criterion_{mn})$$

Finally, in case of ties, the collaborative ranking mechanism sorts the ontologies taking into account not only the average similarity between the ontologies and the evaluations stored in the system, but also the total number of evaluations of each ontology, providing thus more relevance to those ontologies that have been rated more times.

$$\frac{M}{M_{total}} ranking(ontology)$$

7.3.3 Experiments

In this section, we present some small-scale experiments that attempt to measure: a) the gain of efficiency and effectiveness, and the b) increment of users' satisfaction obtained by the use of our system when searching ontologies within a specific domain.

The scenario of the experiments was the following. A repository of thirty ontologies was considered and eighteen subjects participated in the evaluations. They were Computer Science Ph.D. students of our department, all of them with some expertise in modeling and exploitation of ontologies. They were asked to search and evaluate ontologies with our ontology evaluation platform in three different tasks. For each task and each student, one of the following problem domains was selected:

- **Family**. Search for ontologies including family members: mother, father, daughter, son, etc.
- **Genetics**. Search for ontologies containing specific vocabulary of Genetics: genes, proteins, amino acids, etc.
- **Restaurant**. Search for ontologies with vocabulary related to restaurants: food, drinks, waiters, etc.

In the repository, there were six different ontologies related to each of the above domains, and twelve ontologies describing other no related knowledge areas. No information about the domains and the existent ontologies was given to the students.

Tasks 1 and 2 were performed first without the help of the collaborative modules of the system. After all users finished the previous ontology searches and evaluations, task 3 was done with the collaborative components activated. For each task and each student, we measured the time expended, and the number of ontologies retrieved and selected ('reused'). We also asked the users about their satisfaction (in a 1-5 rating scale) about each of the selected ontologies and the collaborative modules.

Table 7.9 and Table 7.10 contain a summary of the results. Note that measures of task 1 are not shown. We have decided not to consider them for evaluation purposes because we discern the first task as a learning process of the use of the tool, and its time executions and number of selected ontologies as skewed no objective measures.

To evaluate the enhancements in terms of efficiency and effectiveness, we present in Table 7.9 the average number of reused ontologies and the average execution times for task 2 and 3. The results show a significant improvement when the collaborative modules of the system were activated. In all the cases, the students made use of evaluations suggested by others, accelerating the processes of problem definition and relevant ontology retrieval.

	Task 2 (without collaborative modules)	Task 3 (with collaborative modules)	% improvement
# reused ontologies	3.45	4.35	26.08
Execution time	9.3	7.1	23.8

Table 7.9 Average number of reused ontologies and execution times (in minutes) for tasks 2 and 3

On the other hand, table 9 shows the average degrees of satisfaction revealed by the users about the retrieved ontologies. Again, the results evidence positive applications of our approach.

Task 2	Task 3	% improvement
3.34	3.56	6.58

Table 7.10 Average satisfactions values (1-5 rating scale) for ontologies reused in tasks 2 and 3.

7.4 Exploiting the semantic knowledge gateway

The implemented SW gateway stores and gives access to multiple ontologies in an efficient way thanks to its indexing structures and ontology-accessing API (section 7.2). While the indexing structures let to efficiently retrieve semantic information from large-scale ontological repositories, the accessing API encapsulates the details of ontology languages, storage types, and locations, allowing applications to easily manage several diverse ontologies at a time.

Built upon the above storage structures and accessing API, the SW gateway also provides a number of ontology evaluation measures and algorithms which aim to select the best available semantic content for users and applications.

In this section, we explain how our ontology-based IR model takes advantage of the implemented storage structures, multi-ontology accessing APIs, and evaluation measures and algorithms to select the most appropriate semantic Web content for the query processing and document annotation modules. We also describe those aspects of such structures and algorithms that should be further included in the modules.

7.4.1 Exploiting the semantic knowledge gateway for query processing

As explained in section 6.2.2, the query processing module PowerAqua makes use of the SW storage and accessing structures, but does not exploit the ontology evaluation measures and algorithms provided by our SW gateway.

The lack of such measures and evaluations is justified when attempting to exploit them within a Web scale retrieval environment. In our semantic retrieval model, the Golden Standard is defined as a natural language query. This query is introduced by the user, and, in the ideal case, he should not refine it or provide additional feedback about his search interests and goals. If we want to keep the usability constraint of using a Web scale search system, human intervention should be minimized as much as possible, restriction which has a negative impact in the evaluation measures previously presented:

- Regarding the *lexical evaluation*, a) the number of words in the query is generally very small to clearly describe the domain of interest, and b) some of the words in the query describe relationships among domain entities but not domain content. A natural language processing of the query could be done to discard these words (e.g. removing prepositions, stop-words, etc.). However, this action might result in a significant loss of expressiveness of the query. For instance, imagine the following two queries: a) "what is the number of Spaniards inside Spain?" and b) "what is the number of Spaniards outside Spain?" If we remove the prepositions "inside" and "outside", both queries lose a significant part of their semantics.
- Regarding the *taxonomical evaluation*, it is clear that a user query does not provide any hierarchy structure. Several tests have been done to automatically expand the query terms using WordNet, without any human supervision. However, the level of ambiguities introduced in the query makes this expansion process unsatisfactory.
- Regarding the *collaborative evaluation*, human intervention is required, although, in this case, it is not necessarily needed at query time.

Therefore, if we want to represent the Golden Standard as a natural language query, the proposed ontology evaluation strategies seem to not be appropriate. The information expressed by natural language queries is not enough to describe a domain of interest, and human intervention is not desirable to refine it.

To address these problems, PowerAqua introduces its own *lexical* and *taxonomical* evaluation measures, allowing an automatic mapping between a user request and its suitable ontologies. These ontology-selection algorithms consider all the expressiveness of natural language queries, and take advantage of WordNet and the SW content as background knowledge to better discriminate the ontologies that can potentially contain an answer to the given request.

On the other hand, aiming to incorporate *collaborative* evaluation measures, it would be interesting, as a future work extension, to complement PowerAqua's ontology selection algorithms with an ontology visualization and evaluation module where the user could assess the ontologies from which the answers to his request are extracted. The user's assessments can then be used by PowerAqua at query time to re-rank the set of ontologies and discern the best semantic knowledge according to user's opinions.

7.4.2 Exploiting the semantic knowledge gateway for annotation

As explained in section 6.2.1.1, the storage and accessing structures of the SW gateway are used by the two proposed annotation processes. However, the ontology evaluation and selection measures were not included in them.

As future work, we plan to develop a third annotation algorithm which, using as Golden Standard the terms extracted from the Web documents, and the proposed lexical evaluation measures, would be able to retrieve the subset of ontologies more likely to contain the domains or topics expressed by the documents. This subset of ontologies would be then used during the annotation process. We hypothesize that reducing the set of ontologies could help to improve the performance, and cut down the annotation ambiguities.

7.5 Discussion

Current ontology-based applications, including ontology-based search systems, are generally limited to specific domain environments. To develop a novel generation of SW applications (Motta & Sabou, 2006) that do not present this close-domain restriction, it is necessary to consider among others three main requirements: a) semantic data reuse vs. generation, b) multi-ontology vs. single-ontology systems, and, c) scale as a important as data quality.

This chapter described the work carried out in this thesis to provide an appropriate SW gateway which allows semantic applications, and more specifically our ontology-based retrieval system, to satisfy the above requirements. The implemented SW gateway lets to:

- **Reach a consensus on the access to the available distributed SW content** by the implementation of a generic API that provides a unique access to the semantic content independently on the ontology description language (OWL, RDFS, DAML, etc.), the ontology storage type (database, file, URLs, etc.), and the ontology accessing framework (Jena, Sesame, etc.).
- **Provide ontology-indexing structures to scale the access to the SW content**. The indices enable applications to access semantic content quickly and efficiently. For cases where more explicit semantic information is needed, these indices can also help to select just the needed subset of semantic content to load in memory, therefore increasing the scalability of applications.

- **Provide ontology evaluation measures that allow users or applications to select the most appropriate SW content**. Two kinds of evaluation measures have been implemented: a) *content based evaluation measures*, which aim to lexically and taxonomically compare the available ontologies with a Golden Standard describing the domain of interest, and, b) *collaborative evaluation measures* that aim to "exploit the wisdom of the crowds" to evaluate the quality of ontologies, and reuse the most appropriate ones according to their subjective quality.

The SW gateway has been integrated with the semantic retrieval model proposed in this thesis. The storage and accessing structures and APIs have been satisfactorily used by both the annotation and query processing modules. However, several improvements and extensions of the ontology evaluation measures should be performed to adapt them to the selection processes demanded by our ontology-based retrieval system.

The main identified problem is related to the Golden Standard definition. The implemented ontology selection algorithms need a Golden Standard description containing lexical and taxonomical information. However, the query and annotation processes of our ontology-based retrieval system do not extract any taxonomical information from queries and documents. While it is feasible to extract some relations using natural language approaches, this improvement has been set up as a future extension. Meantime, our ontology-based retrieval model takes advantage of PowerAqua's ontology selection algorithms to extract the most appropriate semantic information for each individual query.

Chapter 8

Coping with knowledge incompleteness by rank fusion

As already pointed out along this document, semantic knowledge incompleteness is an inherent condition to any attempt at formalizing semantics in any realistic application domain, and is thus an intrinsic assumption in our research. Our solution to this problem is to complement the pure semantic retrieval model with keyword-based techniques. With the aim of retaining keyword-based search recall when the available semantic information is scarce or incomplete, our proposed semantic retrieval model (see chapters 5 and 6), combines in a final ranking list the results obtained by means of our ontology-based retrieval algorithms and a traditional keyword-based search approach. The target of this chapter is therefore to study different techniques of ranking fusion to further enhance the reliability and robustness of the combined retrieval performance. The chapter is divided in three main sections: section 8.1 motivates the problem of ranking combination; section 8.2 provides a brief state of the art of the different ranking combination techniques. Section 8.3 proposes a new approach that takes into consideration statistical information in order to improve the combination of rankings. Finally section 8.4 shows the evaluation of this novel ranking fusion model and a brief discussion of its advantages and contributions.

8.1 Motivation

As observed in chapters 5 and 6 the performance of our proposed model is in direct relation with the amount and quality of information within the ontologies and KBs. The latest studies to characterize the knowledge available in the SW show that, even though the amount of knowledge published on the SW – i.e. the number of ontologies and KBs available online – is rapidly increasing, the SW is still sparse and incomplete (Sabou, Gracia, Angeletou, D'Anquin, & Motta, 2007) (D'Aquin, Gridinoc, Sabou, Angeletou, & Motta, 2007). In consequence, tolerance to incomplete knowledge has been set as an important requirement in our proposal. This means that, in our semantic retrieval approach recall and precision of keyword-based search shall be retained when ontology information is not available or incomplete.

To reach this goal, the semantic retrieval model proposed returns a combined ranking that aggregates the results coming from our ontology-based retrieval model and the results returned by traditional keyword-based techniques[59]. However, the combination of rankings is tricky. While the inclusion of keyword-based results ensures the robustness of our method when ontology-based results are bad, this is at the expense of a precision loss in the opposite case. Achieving an appropriate balance between keyword-based and ontology-based results is essential for the reliability of our semantic retrieval approach. This combination can be seen as a rank aggregation problem. Relevant subproblems include score normalization, score weighting, and measure selection for dynamic weight adjustment, which are addressed in depth in the present chapter.

Rank aggregation has been a widely addressed research topic in the field of Information Retrieval, among others (Aslam & Montague, 2001). Given a set of rankings which apply to a common universe of information objects, the task of rank aggregation consists of combining this list in a way to optimize the performance of the combination. Examples where rank fusion takes place include, for instance, metasearch (Pennock, Horvitz, & Giles, 2000), multi-criteria retrieval (Dasiopoulou, 2005) (e.g., sort books from a bookstore by a combination of topic relevance, price, ratings, delivery time, etc.), distributed search from heterogeneous sources (Berretti, Del Bimbo, & Pala, 2004), personalized retrieval (Castells, et al., 2005), group-based retrieval (Manmatha & Sever, 2002) or classification based on multiple evidence (Aslandogan & Yu, 2000)

The target of this chapter is to study different techniques of rank fusion to further enhance the reliability and robustness of the combination of ontology-based and keyword-based results, and therefore, optimize the performance of our semantic retrieval system. The problem is addressed at different points in the combination process, such as the normalization of the scores returned by ontology-based and keyword-based approaches, and the selection of the weights for the linear combination of both scores. The behavioural patterns of the search engines (drawn from long-term observations) and global properties of the search space are used as further context to drive the combination process.

[59] http://lucene.apache.org/java/docs/

8.2 State of the art in rank fusion

A rank fusion technique can be **characterized** by the input data it requires and the ways it can be used (Montague & Aslam, 2001):

- **Input**: Ng et (Ng & Kantor, 2000) proposed a classification of metasearch techniques based on the input used by the system to combine the list of rankings returned by the different search engines:
  - Decision-level fusion: ranked list of results. The combination is based solely on the position of items in the different rankings.
  - Signal-level fusion: result relevance scores. The combination is based on the scores associated to the different rankings, according to which the items are ordered by each system.
  - Data-level fusion: full documents, or document title + short summary.
- **Training data:** Training data is sometimes available to the aggregation system. It usually consists of relevance judgments manually provided by experts for documents (e.g., with respect to queries). Statistics on the average performance of each input system can also be used. Training data is usually expensive to obtain, so it may not always be available for the fusion technique.
- **Overlap:** The metasearch problem can be studied in the context of:
  - *Data fusion*, where all the systems to be combined are defined on the same collection of information objects.
  - *Collection fusion*, where the inputs of the systems are completely disjoint.
  - *Arbitrarily overlapping* inputs, where the systems operate on different, but not disjoint collections.
- **Application:** The metasearch can be classified as *external*, if it uses complete search systems (seen as black boxes), and tries to improve them by combining their results, or *internal*, if it is at heart of a retrieval system where different subsystems collect evidence from several sources (e.g., multimodal ones) or different criteria (e.g., number vs. proximity of query keyword occurrences in text documents, structural properties, link analysis, etc.).

Fusion techniques for metasearch typically bring the following **benefits** (Aslam & Montague, 2001):

- **Better recall:** Recall is the ratio of retrieved relevant documents to total relevant documents. In the case of different collections, the recall improvement is clear, since the relevant documents that are missing in a collection are never returned by the corresponding system. In the case of identical collections, a recall improvement may result from different systems returning different documents. However, Lee (Lee J. H., 97) argues that this improvement is often marginal because of the systems return many different irrelevant documents but many same relevant documents.
- **Better precision:** Precision is the ratio of retrieved relevant documents to retrieved documents. Saracevic and Kantor (Saracevic & Kantor, 1998) show how the probability of a

relevant document being retrieved increases monotonically with the number of search engines that retrieve it.

- **Consistent performance:** A single search engine may give different answers for the same query over time (this is characteristic of e.g., Web search engines). Since the fusion techniques combine many different results each time, the resulting instability is smaller, enhancing the consistency of the global system (Montague & Aslam, 2001).

Interestingly, Manmatha and Sever (Manmatha & Sever, 2002) observed that combining more than 5 engines do not seem to bring a substantial improvement in performance, and may in fact cause degradation. However, this observation has not been found in other studies.

The metasearch problem can be decomposed into three main subproblems (Montague & Aslam, 2001):

- **Normalization:** In order to combine different rankings, the outputs have to be first made comparable across systems. Normalization is needed for both rank-based and score-based fusion systems. Regarding the latter, the scores returned by the different information retrieval systems may not be equivalent, e.g., they have different scales, different ranges, and completely different distributions. Regarding the former, the rank lists returned by different input systems may have different lengths. Also, care must be taken of the rank estimation for items that do not appear in all the lists. The normalization techniques can also use training information in order to make the system outputs more equivalent.
- **Estimation:** Typically, not just the order of items varies across the input systems, but also the set of items in the rankings is not the same. The estimation problem refers to assigning an estimated score to a document for a system that has not returned it.
- **Combination:** This is the operation that has received most attention in the literature. It refers to using the normalized information returned by the different input systems to combine all the results in a unique output list.

The relevant state of the art related to the above subproblems is summarized in the next two sections. The following **notation** will be used in the sequel.

Ω The universe of information objects to be ranked.

P The set of rank sources to be combined.

τ A rank source $\tau \in P$.

Ω_τ The set of items $\Omega_\tau \subset \Omega$ returned by τ.

$\Omega_R = \bigcup_{\tau \in R} \Omega_\tau$ The set of all items $\Omega_P \subset \Omega$ returned by at least one source in P.

$\tau(x)$ Given $x \in \Omega_\tau$, $\tau(x)$ is the position of x in the ranking returned by τ.

$s_\tau(x)$ The score assigned to x in the ranking returned by τ.

$\overline{s}_\tau(x)$ The normalized score for x corresponding to τ.

$s_P(x)$ The final combined score for x.

8.2.1 The normalization and estimation problem

Normalization techniques can be divided into different categories, depending on whether they apply to scores or rank positions, and whether training data is used to improve the process. The following table summarizes the main approaches described in the literature.

	Training data	
	No	Yes
Score-based	Standard Sum ZMUV 2MUV Manmatha	
Rank-based	Rank-sim Borda	Bayes

Table 8.1 Normalization techniques

Score-based normalization shall be discussed first, after which a description of rank-based methods will follow.

In order to describe score normalization, it is important to stress that relevance scores are real functions $s_\tau : \Omega_\tau \to P$, where the score $s_\tau(x)$ is used to define the position of x in the list of documents returned by τ, in a way that if $s_\tau(x) < s_\tau(y)$ then y is before x in the ranking, but we cannot make any further assumption about s_τ (Montague & Aslam, 2001).

The so-called **standard** score normalization method is as follows (see e.g., (Lee J. H., 97)):

$$\overline{s}_\tau(x) = \frac{s_\tau(x) - \min_{y\in\Omega_\tau} s_\tau(y)}{\max_{y\in\Omega_\tau} s_\tau(y) - \min_{y\in\Omega_\tau} s_\tau(y)}$$

This method scales all the scores to the interval [0,1]. For the estimation problem, any item not retrieved by a system τ is assigned a normalized score $\overline{s}_\tau(x) = 0$.

Montague and Aslam (Montague & Aslam, 2001) proposed the Sum, ZMUV and 2MUV score normalization methods, based on further requirements for the normalization scheme, namely:

- *Shift-invariant*: normalization should be insensitive to input shifts.
- *Scale-invariant*: normalization should be insensitive to scaling by a multiplicative constant.
- *Outlier-insensitive*: normalization should not be overly sensitive to the score of a single document.

The following table summarizes the properties of the different normalization techniques.

	Sum	ZMUV	2MUV
Description	Shift min to 0 Scale sum to 1	Shift mean to 0 Scale variance to 1	Shift mean to 2 Scale variance to 1
Shift-invariant	Yes	Yes	Yes
Scale-invariant	Yes	Yes	Yes
Outlier-insensitive	Sensitive to the min score only	Yes	Yes

Table 8.2 Score normalization methods proposed by Montague and Aslam[60].

- The **Sum** method scales the minimum value to 0 and the sum of all scores returned by the system to 1. This method is shift- and scale-invariant and is sensitive only to the minimum score given for each query. In order to solve the estimation problem, again, any item that is not retrieved by the system is assigned a normalized relevance score of 0.

- The **ZMUV** (Zero Mean, Unit Variance) method scales the mean to 0 and the variance to 1. This method is shift- and scale-invariant, but also outlier-insensitive because it does not depend directly on the min or max scores of the returned collection. In order to solve the estimation problem, any document that was not retrieved by the system is assigned a normalized relevance score of –2, in order to maintain the average in 0.

- The **2MUV** method is a variant of ZMUV that shifts the mean to 2 instead of 0. This forces most scores to be positive, which is needed e.g., in order to use this type of normalization e.g., with the CombMNZ and CombANZ combination techniques (Fox & Shaw, 1993), which will be described in the next section.

The experiments reported in (Montague & Aslam, 2001) show that normalization schemes not sensitive to outliers yield better performance. The authors predict that a better score estimation for non retrieved documents may also increase the performance of the ranking fusion methods, but this is still an open problem.

Manmatha (Manmatha & Sever, 2002) provides a theoretical justification to the Sum and ZMUV normalization schemes. Moreover, he analyzes the probabilistic behaviour of search engines, in order to derive a better combination of their outputs (Manmatha, Rath, & Feng, 2001) (Manmatha & Sever, 2002). It is observed that the scores typically follow an exponential distribution for the set of non-relevant documents, and a Gaussian distribution for the set of relevant ones. According to this, a score $s_\tau(x)$ output by a given engine τ for a document x is normalized to $\bar{s}_\tau(x) = \mathrm{P}\left(y \text{ is relevant} \mid s_\tau(y) = s_\tau(x)\right)$, where y is randomly chosen in Ω. The conditional relevance probability is computed easily by applying Bayes' rule:

[60] (Montague & Aslam, 2001)

$$\overline{s}_\tau(x) = \frac{\mathrm{P}\left(s_\tau(y) = s_\tau(x) \mid y \text{ is relevant}\right)\mathrm{P}\left(y \text{ is relevant}\right)}{\begin{array}{c}\mathrm{P}\left(s_\tau(y) = s_\tau(x) \mid y \text{ is relevant}\right)\mathrm{P}\left(y \text{ is relevant}\right)\\ +\mathrm{P}\left(s_\tau(y) = s_\tau(x) \mid y \text{ is not relevant}\right)\mathrm{P}\left(y \text{ is not relevant}\right)\end{array}}$$

P ($s_\tau(y) = s_\tau(x) \mid y$ is relevant) and P ($s_\tau(y) = s_\tau(x) \mid y$ is not relevant) are approximated by a Gaussian and an exponential distribution respectively. The density parameters (mean and standard deviation) in the density functions that define these probabilities, as well as the probabilities P (*x* is relevant) and P (*x* is not relevant), are approximated by modeling the density function $f(s_\tau(y) = s_\tau(x))$ as a mixture of an exponential and a Gaussian relevance distribution using the Expectation Maximization method.

Regarding rank-based methods, they are based solely on the position $\tau(x)$ returned for *x* by each input system, which is an integer value ranging between 1 and the number of items returned by the input system. Rank-based methods have the major advantage that the score values are not always made available by input systems (take e.g., typical Web search engines).

One of the first rank-based strategies, **Rank-sim**, was proposed by Lee (Lee J. H., 97):

$$\overline{s}_\tau(x) = 1 - \frac{\tau(x) - 1}{|\Omega_\tau|}$$

Lee compared the performance of this method against standard score normalization. Despite the fact that rank-based normalization is not sensitive to outliers, the experimental results have shown that the standard score-based technique slightly outperforms Rank-sim, except when the input systems have quite different similarity / rank curves (Lee J. H., 97).

The **Borda** method (Aslam & Montague, 2001) is based on the Borda Count voting method where each search engine is seen as a voter. Each voter ranks a fixed number *c* of candidates, where the first candidate is given a score of *c*, the second *c* – 1, and so on. If there are any candidates left unranked by the voter, the remaining points are evenly distributed among them:

$$\overline{s}_\tau(x) = \begin{cases} 1 - \dfrac{\tau(x) - 1}{|\Omega|} & \text{if } x \in \Omega_\tau \\ \dfrac{|\Omega| - |\Omega_\tau| + 1}{2|\Omega|} & \text{otherwise} \end{cases}$$

The **Bayes** method (Aslam & Montague, 2001) is based on Bayesian inference. The score for an item *x* in a system τ is computed by estimating the probability that a relevant document be ranked at the position $\tau(x)$, and the probability that an irrelevant document be ranked at $\tau(x)$, as follows:

$$\overline{s}_\tau(x) = \log \frac{\mathrm{P}\left(\tau(x) \mid x \text{ is relevant}\right)}{\mathrm{P}\left(\tau(x) \mid x \text{ is not relevant}\right)}$$

P(τ(x) | x is relevant) and P(τ(x) | x is not relevant) are approximated by using the TREC eval program (Voorhees & Harman, 2000) to compute the average precision at different rank levels, and the average documents per query. Alternative ways to estimate the probabilities, like smoothing and interpolating, performed worse.

8.2.2 The combination problem

Again, the combination problem can be divided into two main categories depending on whether the combination takes as input the (normalized) scores or just the ordered lists. The distinction between techniques using and not using training data is useful as well.

	Training data	
	No	**Yes**
Score-based	CombMIN CombMAX CombSUM CombANZ CombMNZ	Bartell Vogt
Rank-based	Markov chain Borda Weighted Borda	Bayes
Hybrid		Logistic regression

Table 8.3 Combination techniques

Fox and Shaw (Shaw & Fox, 1994) describe some of the most popular and effective combination algorithms to date, which are score-based:

- **CombMIN:** $s_{\mathrm{R}}\left(x\right)=\min_{\tau\in\mathrm{R}}\bar{s}_{\tau}\left(x\right)$.
- **CombMED:** $s_{\mathrm{R}}\left(x\right)=\operatorname{median}_{\tau\in\mathrm{R}}\bar{s}_{\tau}\left(x\right)$.
- **CombMAX:** $s_{\mathrm{R}}\left(x\right)=\max_{\tau\in\mathrm{R}}\bar{s}_{\tau}\left(x\right)$.
- **CombSUM:** $s_{\mathrm{R}}\left(x\right)=\sum_{\tau\in\mathrm{R}}\bar{s}_{\tau}\left(x\right)$.
- **CombANZ:** $s_{\mathrm{R}}\left(x\right)=\frac{1}{\mathrm{h}\left(x,\mathrm{R}\right)}\sum_{\tau\in\mathrm{R}}\bar{s}_{\tau}\left(x\right)$, where h($x$,P) is the number of input systems that retrieve x.
- **CombMNZ:** $s_{\mathrm{R}}\left(x\right)=\mathrm{h}\left(x,\mathrm{R}\right)\sum_{\tau\in\mathrm{R}}\bar{s}_{\tau}\left(x\right)$.

According to the experiments reported in (Fox & Shaw, 1993) and (Lee J. H., 97), CombMNZ is considered to be the best method, even though it performs just slightly better than CombSUM. CombMNZ is motivated by the observations by Lee regarding the overlap between the relevant and not relevant documents retrieved by different search engines: "*Different runs retrieved similar sets of relevant documents but different set of non relevant documents*" (Lee J. H., 97).This unequal overlap property also coincides with Kantor's results: "*The more runs a documents is retrieved by, the higher the rank that should be assigned to the document*" (Saracevic & Kantor, 1998). The results obtained by Fox and Shaw have been later improved by further elaborating on the normalization step (Lee J. H., 97) (Montague & Aslam, 2001).

In the definition of the combination methods given above, all the input systems are given the same priority. Bartell (Bartell, 1994) and Vogt (Vogt & Cottrell, 1999) propose a variant of CombSUM consisting of the introduction of a weight α_τ for each source, according to the importance, quality, reliability, etc., of the sources, so that the fused score is computed as a weighted linear combination $s_{\mathrm{R}}(x) = \sum_{\tau\in\mathrm{R}} \alpha_\tau \cdot \overline{s}_\tau(x)$. Of course, this approach can be applied to CombMNZ and CombANZ as well.

Bartell (Bartell, 1994) describes different strategies to set the weights α_τ, such as the *Conjugate Gradient Method*, which uses the *Guttman's Point Alienation* statistic as a target function to maximize. Vogt (Vogt & Cottrell, 1999) (Vogt & Cottrell, 1998) (Vogt, Cottrell, Belew, & Bartell, 1996) extends this by further experiments with linear combination and neural net fusion methods. He observes that the best results are obtained for queries with many relevant documents, and very different input systems. These models require training data to set the weights for the linear combination.

Score-based combination can also be combined with rank-based normalization. For instance, the **Bayes** normalization followed by CombSUM is competitive with score-based techniques (Aslam & Montague, 2001). The performance can significantly exceed the best existing strategy when combining the results of disparate systems.

In the basic the **Borda** method, the normalized scores (votes) computed for each source (as described in the previous section) are summed for each item, and the items are ranked by the total points obtained (i.e. the candidate with more points wins the election) (Aslam & Montague, 2001), which is in fact equivalent to the CombSUM strategy. In the context of metasearch, it is not always clear that each voter should have the same importance. A variant of this technique is **weighted Borda**, in which votes are weighted taking into account the quality of the source. Bartell (Bartell, 1994) and Vogt (Vogt & Cottrell, 1999) compute these weights by assessing the performance of different search engines using available training data; however these techniques have not proved to improve performance.

Savoy (Savoy, Le Calvé, & Vrajitoru, 1996) proposes a hybrid rank-based and score-based approach, where ranks and scores are combined in a **logistic regression** model, as follows:

$$s_{\mathrm{R}}(x) = \frac{1}{1+e^{-\alpha-\beta\cdot u(x)}}, \text{ where } \beta\cdot u(x) = \sum_{\tau\in\mathrm{R}} \beta_{\tau,1}\cdot\tau(x) + \beta_{\tau,2}\cdot s_\tau(x) + \beta_{\tau,3}\cdot \mathrm{var}_\tau(x)$$

In the above formula α, $\beta_{\tau,1}$, $\beta_{\tau,2}$, $\beta_{\tau,3}$, are parameters to be learnt for each source, and var_τ is the variance of the normalized relevance scores for the source τ. The results obtained by this approach are slightly superior to the linear combination schemes.

As to purely rank-based combination methods, a good performance has been achieved based on **Markov chain** models, where the set of states is Ω_P, and the transition matrix M is computed by different strategies, based on the rankings P produced by the different input systems (Dwork, Kumar, Noar, & Sivakumar, 2001). Some specific models to define the transitions are:

- **MC$_1$:** From a current state $x\in\Omega_P$, choose uniformly form all the sources a state $y\in\Omega_P$ with a higher rank $\tau(y) \geq \tau(x)$ than the current state for some source $\tau\in P$.
- **MC$_2$:** From a current state $x\in\Omega_P$, first choose uniformly one source $\tau\in P$ so that $x\in\Omega_\tau$, and then take a state $y\in\Omega_\tau$ with a higher rank $\tau(y) \geq \tau(x)$ than the current state for this τ.
- **MC$_3$:** Equivalent to MC$_2$ but choosing uniformly a state $y\in\Omega_\tau$. If $\tau(y) \geq \tau(x)$, then the next state is y, and otherwise we stay in x.
- **MC$_4$:** From a current state $x\in\Omega_P$, choose uniformly a state $y\in\Omega_P$ If $\tau(y) \geq \tau(x)$ for the majority of $\tau\in P$ with both $x\in\Omega_\tau$, $y\in\Omega_\tau$, then move to y, else stay in x.

The probability values for each state in the stationary distribution of the Markov chains defined by these models is taken as the score that determines the fused ranking. In other words, if P : $\Omega_P \rightarrow [0,1]$ is a stationary distribution, then $s_P(x) = P(x)$. Renda and Straccia (Renda & Straccia, 2003) show by several comparative experiments that, contrary to what is generally assumed, the performance of these rank-based methods is comparable to the score-based. They also show that although there is not a clear winner between the different Markov models, MC$_1$ and MC$_4$ tend to be the top ones

8.3 Optimizing rank fusion: proposed approach

As introduced in the previous section, rank fusion is a pervading operation in applied information retrieval technology (Montague & Aslam, 2001). To name a few examples, rank aggregation takes place in the combination of multiple criteria and heuristics for document/query similarity assessment in most search engines; in merging the outputs of different engines in meta-search tools; in the combination of query-based and preference-based relevance scores for personalized search or even in the combination of preferences from multiple users in group-aware systems.

In this thesis, rank fusion is identified as a critical and delicate operation for the combination of two different ranking criteria (see chapters 5 and 6), the first one from the perspective of reliability of the standard keyword-based search approaches, and the second one, from the unpredictable performance of ontology-based retrieval algorithms.

This section addresses two fundamental problems in the combination of relevance scores: the normalization of the rank sources, and the combination of the normalized rankings. 8.3.2 describes a general technique for the optimization of rank fusion strategies at the normalization phase. The technique does not make any assumption about the nature of the rank sources or the purpose of their combination, and can be therefore used in other contexts beyond semantic search. 8.3.3 discusses how to optimize the performance and reliability of semantic retrieval in the combination of normalized keyword-based and ontology-based scores, analyzing the problems involved and proposing several techniques to address them.

8.3.1 Score normalization

Prior research on rank fusion has explored both rank-based and score-based aggregation techniques (Renda & Straccia, 2003). In either case, the rankings have to be made comparable before they are combined (Croft, 2000). For example, the 10th position in the ranking has a quite different meaning when 15 results are returned than it would within 1,000 results. Similarly, a score of 0.9 has not the same meaning in a system ranging in [0,1] as in one ranging in [0,100]. In this section we propose a general-purpose enhancement for the normalization step that is applicable to any score-based combination technique.

In general, score-based techniques tend to perform better than rank-based ones (Lee J. H., 97), which can be explained by the fact that scores carry more information than the rank position: the latter can be obtained from the former, but the opposite is not true. At the same time, and for the same reason, we should note that the information contained in score values is more difficult to uniformize across systems (the entropy, i.e. the potential heterogeneity of measures increases with the amount of information in the measures).

Indeed, score-based techniques may be sensitive to artificial deviations occurring consistently in the individual score distributions, which do not affect at all the result of each ranking technique separately, but distort the combined result when the individual biases differ from each other, and therefore it should be possible to improve the results by undoing these deviations. This was already noted by Manmatha et al (Manmatha, Rath, & Feng, 2001), among other authors. For instance, one technique may score items on a logarithmic scale, while others do so on a linear scale, a quadratic scale, etc., so that normalizing the scoring range linearly to the same interval is not enough for a consistent combination of score values. This is especially true in systems, where the scoring techniques are defined based on different media (images, ontologies, keywords), and really mean different things.

In order to devise a general method to merge the output of several ranking techniques, it is not possible to make any a-priori assumption on the interpretation of the scores values. The values may correspond to a degree of relevance, probability of relevance, odds of relevance, or other interpretations in a variety of retrieval models, often undergoing further mathematical transformations (scaling, dampening, logs, etc.) for practical purposes. The only assumptions that can be made on the scoring functions are that they return values in P, and they induce a partial order on the set of information objects that approximates as accurately as possible the order of relevance. However, as pointed out before, in order to combine the scores, the values should be first made comparable across input systems, which usually involves a normalization step (Montague & Aslam, 2001).

Instead of a simple normalization based on linear transformations (as the ones studied in (Lee J. H., 97)), and in order to compensate for the biases in the input scorings, we propose an aggregation model where the source scores are normalized to a common *ideal* score distribution, and then merged by a linear combination. Some experiments on available data from several TREC collections have been carried out to validate our proposal.

The details of the proposed normalization technique are given in the next subsection. After that, section 8.3.3 explains how the score distributions can be estimated in a practical application, and section 8.4 shows the experimental results using several TREC collections.

8.3.2 Normalization of score distributions

In prior work, normalization typically consists of linear transformations (Lee J. H., 97) and other relatively straightforward, yet effective methods, such as normalizing the sum of scores (rather than the max) of each input system to 1, or shifting the mean of values to 0 and scaling the variance to 1 (Montague & Aslam, 2001). But none of these strategies takes into account the detailed distribution of the scorings, and is thus sensitive to "noise" score biases.

As explained earlier in section 8.2 on the state of the art, a work where the score distribution is taken into account is that of Manmatha et al (Manmatha, Rath, & Feng, 2001), who analyze the probabilistic behaviour of search engines, in order to derive a better combination of their outputs. They observe that the scoring values have an exponential distribution for the set of non-relevant documents, and a Gaussian distribution for the set of relevant ones. According to this, a score *s* output by a given engine for a document *x* is normalized to P (*x* is relevant | *score*(*x*) = *s*). This probability is computed easily by applying Bayes' rule and approximating P (*x* is relevant) and P (*x* is not relevant) by a mixture of an exponential and Gaussian relevance distribution using the Expectation Maximization method (Manmatha & Sever, 2002).

Starting from Manmatha's analysis of typical score distributions, we propose an alternative approach, where input scores are actually mapped to an *optimal score distribution* (OSD), which we define as an approximation to the score distribution of an ideal scoring function that matches the ranking by *actual relevance*. Of course this is a difficult concept to define, let alone to obtain, but we claim that an acceptable approximation can provide good results. Before we discuss this, we give an overall outline of our method.

Following the notation introduced in section 8.2, let Ω be the universe of information objects (e.g., documents) to be ranked, and P the set of rank sources to be combined. Each rank source $\tau \in \mathrm{P}$ can be represented as a bijection $\tau : \Omega_\tau \to \mathbf{N}^{+}_{|\Omega_\tau|}$ for some $\Omega_\tau \subset \Omega$, where for each $x \in \Omega_\tau$, $\tau(x)$ is the position of x in the ranking returned by τ (note that τ is not necessarily a total order in Ω, but on a subset $\Omega_\tau \subset \Omega$). For each $\tau \in \mathrm{P}$, we shall denote by $s_\tau : \Omega \to \mathrm{P}$ the scoring function associated to τ, where we take $s_\tau(x) = 0$ if $x \notin \Omega_\tau$ (i.e. x is not "returned" by τ).

Our approach has a static step, in which the appropriate distributions are computed, and a dynamic step, where the outputs of the rank sources are normalized and merged. The offline step is as follows:

1. Build a strictly increasing OSD $\overline{\mathrm{F}} : [0,1] \to [0,1]$. This step is discussed below.

2. For each source $\tau \in \mathrm{P}$, compute the cumulative score distribution F_τ of the values returned by s_τ. This may be approximated by running a significant number of calls to each system with random inputs (e.g., queries and ACEs), as will be described in the next section.

In the dynamic phase, given $x \in \Omega$, let $\{s_\tau(x)\}_{\tau \in \mathrm{P}}$ be the list of scores to be merged. The combination is achieved as follows:

3. Normalization: map the score of each rank to the OSD:

$$s_\tau(x) \rightarrow \bar{s}_\tau(x) = \bar{\mathrm{F}}^{-1} \circ \mathrm{F}_\tau \circ s_\tau(x)$$

4. Combination: merge the normalized scores by a linear combination:

$$s_{\mathrm{R}}(x) = \sum_{\tau \in \mathrm{R}} \alpha_\tau \cdot \bar{s}_\tau(x), \text{ where } \sum_{\tau \in \mathrm{R}} \alpha_\tau = 1$$

The idea of step 3 is illustrated in Fig 8.1. It can be seen that the normalization in this step respects the order of each rank list (except in intervals where F_τ is constant, i.e. where by definition it is unlikely that any score value should fall), since $\bar{\mathrm{F}}^{-1} \circ \mathrm{F}_\tau$ is monotonically non-decreasing. The resulting scores $\bar{s}_\tau = \bar{\mathrm{F}}^{-1} \circ \mathrm{F}_\tau \circ s_\tau$ range in [0,1] (i.e. they are normalized), and their distribution is $\bar{\mathrm{F}}$ for all $\tau \in \mathrm{P}$, thus undoing potential distributional biases, as intended.

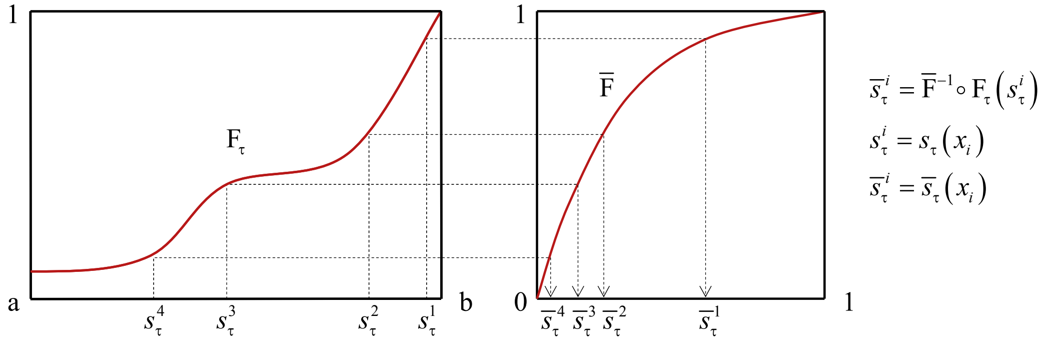


Fig 8.1 Mapping scores to a common standard distribution

The choice of $\bar{\mathrm{F}}$ as an appropriate OSD is critical to our method. One way to approach the problem would be to study the distribution of ratings by actual relevance, where relevance (which is well-known to be itself an elusive notion (Ellis, 1996) (Mizzaro S. , 1997) (Mizzaro S. , 1998)) should be measured on the same continuous scale along [0,1] as the scoring inputs. This approach would build on the hypothesis that on average, good scorings should have a very similar distribution to that of relevance. This principle was followed in such as fundamental work in IR as (Singhal, Buckley, & Mitra, 1996), for a related problem (normalize relevance scores w.r.t. document length) though in a slightly different perspective, where the distribution of relevant documents with respect to the document length was measured (instead of the distribution of scores with respect to documents), using relevance judgements available in TREC collections. This distribution was taken as a target to define optimal similarity functions for retrieval, i.e. relevance assessment by an optimal function should yield the same distribution as the one based on human judgement. A similar strategy could be used for our purpose, but we propose an alternative, inverse approach, namely, that an average distribution of several good scoring systems may provide a rough, though acceptable approximation to an actual relevance distribution. This has the major advantage that relevance information is not needed, e.g., an estimation can be obtained empirically on a statistically significant sample of scoring systems and input values, such as the ones available from the TREC collections. It can also be built by sampling from the rank sources themselves (linearly normalized to [0,1]). In the next section we show how $\overline{F}$ can be approximated this way. In any case, this step is modular in our method, and open to further improvements and research.

8.3.3 Obtaining score distributions

The score distribution F_τ of each input system $\tau \in P$ is computed by observing the behaviour of the ranking system behind each τ (e.g., the keyword-based search module and the ontology-based search module). This means that F_τ is not approximated based only on the scores of τ (i.e. a single ranked result list), but rather by collecting a number of lists of output scores from the ranking system (the one that outputs τ) over several runs, with different input (e.g., queries). From the collected score values, the distribution is approximated based on the histogram for the scores. The more data are collected, the more precise and statistically significant is the histogram, and a better approximation of F_τ is obtained. We have observed in our experiments that in practise, after a fair number of runs the histogram stabilizes and it is not necessary to keep storing information. Figure 8 shows the histogram built for the data set of one of the search engines available in the TREC 8 collection, and the cumulative distribution F_τ computed for this engine.

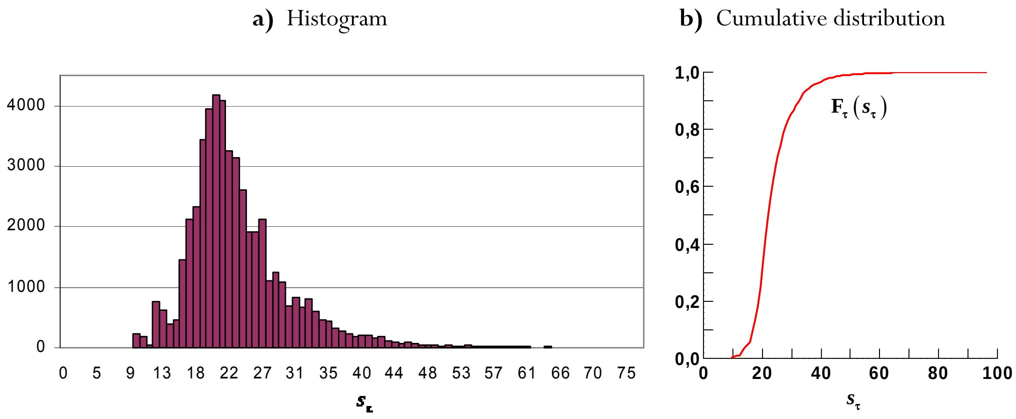

a) Histogram **b)** Cumulative distribution

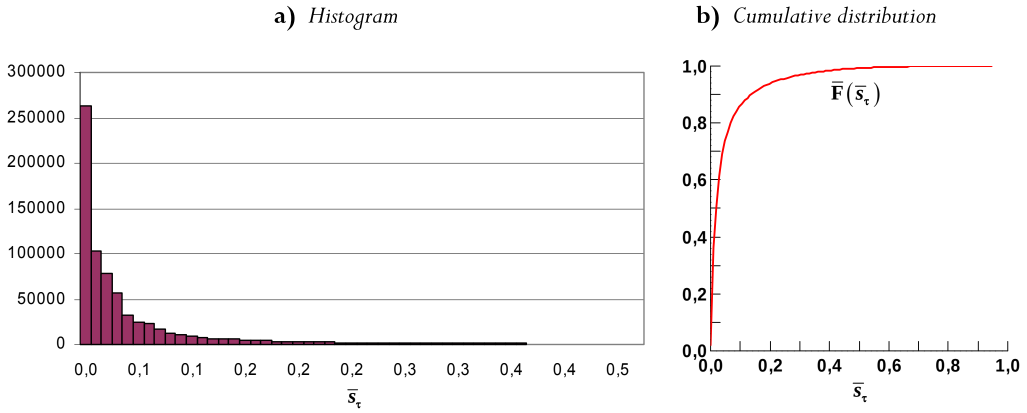
Fig 8.2 Example of the distribution F_τ obtained for the “mds08w1” system in TREC 8

The collected scores from all the systems behind P are used to approximate the OSD $\overline{F}$ as well. To compute $\overline{F}$, the input scores of each system, collected as described before, are first linearly normalized to [0,1] by a variation of the standard score normalization technique described in section 8.2.1, where rather than taking the min and max scores of a single ranked list, all the scores collected from the system over several runs are included. For instance, assume that we wish to merge search engines that take a query as input. For the following explanation, let us use the symbol τ to denote a ranking system (i.e. a list of lists) rather than a single ranked list, and let P denote the set of such systems, the output of which we whish to merge. For each engine τ and each query q with search results available from τ, we would have a ranking τ_q. If Θ is the set of all queries used in the different runs with all the systems, and $\Theta_\tau \subset \Theta$ denotes the subset of queries for which results from τ have been collected, given $q \in \Theta_\tau$ we would normalize the scores of τ_q by:

$$\bar{s}_{\tau_q}(x) = \frac{s_{\tau_q}(x) - \min\limits_{p \in Q_\tau, y \in \Omega_{\tau_p}} s_{\tau_p}(y)}{\max\limits_{p \in Q_\tau, y \in \Omega_{\tau_p}} s_{\tau_p}(y) - \min\limits_{p \in Q_\tau, y \in \Omega_{\tau_p}} s_{\tau_p}(y)}$$

Note that normalizing the scores s_{τ_q} is only needed to compute the OSD $\bar{\mathrm{F}}$, but not F_τ. Note also that for this pre-computation step, the scores are normalized using a method (based on the standard method, described in section 8.2.1) which is different from the distribution-based normalization method that we propose at runtime.

Now the list[61] of all the values for $\bar{s}_{\tau_q}(x)$, for all $\tau \in \mathrm{P}$, $q \in \Theta_\tau$, and $x \in \Omega_{\tau_q}$, is used to build a joint histogram, and the OSD $\bar{\mathrm{F}}$ is defined by linear interpolation of the histogram. Fig 8.3 shows an example where a histogram and a distribution $\bar{\mathrm{F}}$ are built from a set of search engines over four collections from the TREC conferences (the same data set that is used in the experiments that will be presented in the next section).


a) *Histogram*

b) *Cumulative distribution*

Fig 8.3 OSD obtained from the twelve best input systems for TREC 8, 9, 9L and 2001- The graphs show a) the histogram and b) the cumulative distribution of the scores, taking the values $\bar{s}_{\tau_q}$ for all τq, and using a standard score normalization

In our tests, we have observed that OSDs typically have the shape of an exponential distribution, as can be seen in Fig 8.3, and as already found by Manmatha (Manmatha, Rath, & Feng, 2001). Therefore, it would also reasonable to approximate the distribution by $\bar{\mathrm{F}}(s) = 1 - e^{-\lambda s}$, using a maximum likelihood fit. However, in our experiments this approximation does not pass the chi-square

[61] Note that this is a list rather than a set because values may be repeated.

test, and in fact the performance resulting from this approach is slightly inferior to the one achieved from the raw histogram.

Note also that in our approach the statistic data used to build $\overline{\mathrm{F}}$ is being taken from the rank sources themselves to be combined, after a certain period of data collection. Our technique can start to work since the first query, when the only available data is the result set for this query, which enables a very rough, though usually quite acceptable in practice, approximation of $\overline{\mathrm{F}}$. From there on, our system keeps updating $\overline{\mathrm{F}}$ with increasingly better approximations as more queries are answered, until a stable one is reached and data collection can stop. A generic $\overline{\mathrm{F}}$ could also be created just once from some standard collection, and be used elsewhere, so that different engines would be used for the training and the merging. In any case, it should be stressed that our method does not need any relevance information, but only historical scoring data from the input systems.

The use of several input system runs to build F_τ and $\overline{\mathrm{F}}$ is an important difference with respect to the Manmatha approach (Manmatha, Rath, & Feng, 2001) and all other rank aggregation techniques that do not use training data, where only a single result list for each input system to be merged is used. Yet the overhead of handling the extra data in our approach is minimum, as it can be obtained at low cost from the systems to be merged themselves. In particular, with respect to the techniques that do use training data, our approach differs in that it does not use relevance information, which is a major advantage because of the cost of obtaining this information.

8.4 Evaluation and results

We have tested our techniques in four different test collections from the Text Retrieval conferences (TREC), namely TREC 8, TREC 9, TREC 9L, and TREC 2001. For our experiments we have approximated F_τ and $\overline{\mathrm{F}}$ as explained in the previous section, using the scoring data from the TREC collections. For the comparative evaluation we have tried our technique with two reference combination functions after the normalization step that we will name as:

a) **DCombSUM**, where $s_{\mathrm{R}}(x) = \sum_{\tau \in \mathrm{R}} \alpha_\tau \bar{s}_\tau(x)$, i.e. our score normalization step followed by CombSUM (described in section 8.2.2).

b) **DCombMNZ**, where $s_{\mathrm{R}}(x) = \mathrm{h}(x, \mathrm{R}) \sum_{\tau \in \mathrm{R}} \alpha_\tau \bar{s}_\tau(x)$, and h($x$,P) = $\left|\left\{\tau \in \mathrm{R} \mid s_\tau(x) > 0\right\}\right|$ is the number of engines that return x as relevant, a technique used in prior work (Renda & Straccia, 2003), and described as CombMNZ in section 8.2.2.

In both cases we take the same $\alpha_\tau = \dfrac{1}{|\mathrm{R}|}$ for all τ. Note that what is intended to be evaluated is the normalization step, not the combination thereafter.

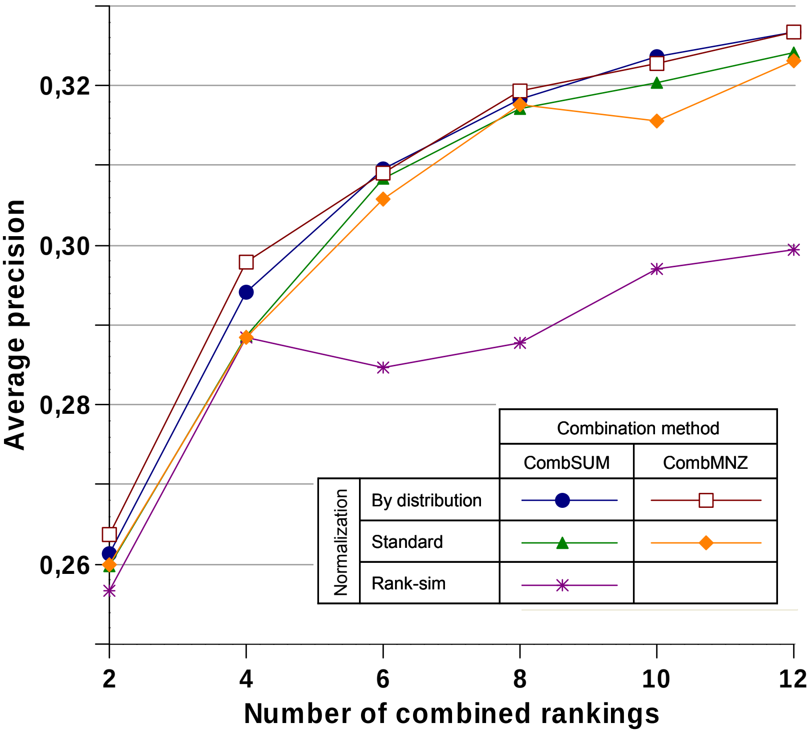


Fig 8.4 Average performance over the four TREC collections

Fig 8.4 show average results over the four collections, and tables 8.4 to 8.9 show the detailed results for each collection, the different runs and the averages. In order to have a fair comparison, we reproduced the same experiments as Renda and Straccia (Renda & Straccia, 2003). We followed their setup, where the top 12 performing rank lists were selected for the experiment (taking no more than one list per search engine), and the tests consist in merging 2, 4, 6, 8, 10, and 12 randomly selected lists, repeating this 10 times, and computing the average precision for each rank aggregation method. As a benchmark, we have chosen the best four aggregation algorithms from (Renda & Straccia, 2003), which we label as MC_4, SCombMNZ, SCombSUM and RCombSUM.[62] MC_4 is one of the Markov Chain combination strategies explained in section 8.2.2. SCombMNZ and SCombSUM correspond to CombMNZ and CombSUM respectively, both using standard score-normalization. RCombSUM corresponds to CombSUM, using the Rank-sim normalization.

[62] SCombMNZ, SCombSum, and RCombSum correspond to $\sum$.s.1, $\sum$.s.0, and $\sum$.r.0 in the original authors' notation in (Renda & Straccia, 2003).

	2	4	6	8	10	12	Avg
MC$_4$	0.3642	0.3627	0.3903	0.3986	0.3999	0.3994	0.3859
SCombMNZ	0.3418	0.3674	0.3852	0.3886	0.3914	0.3943	0.3781
SCombSUM	0.3317	0.3615	0.3746	0.3948	0.3987	0.4028	0.3774
RCombSUM	0.3341	0.3581	0.3558	0.3538	0.3625	0.3625	0.3545
DCombSUM	0.3339	0.3717	0.3935	0.3966	0.4034	0.4082	0.3846
DCombMNZ	0.3342	0.3711	0.3899	0.3996	0.4062	0.4082	0.3848

Table 8.4 Experimental results for TREC 8

	2	4	6	8	10	12	Avg
MC$_4$	0.1486	0.1993	0.2110	0.2093	0.2080	0.2065	0.1971
SCombMNZ	0.1555	0.1730	0.1881	0.1952	0.1966	0.2017	0.1850
SCombSUM	0.1662	0.1854	0.1975	0.2051	0.2066	0.2049	0.1943
RCombSUM	0.1505	0.1644	0.1615	0.1651	0.1740	0.1777	0.1655
DCombSUM	0.1732	0.1887	0.1930	0.2021	0.2080	0.2084	0.1956
DCombMNZ	0.1704	0.1918	0.1996	0.2018	0.2028	0.2084	0.1958

Table 8.5 Experimental results for TREC 9

	2	4	6	8	10	12	Avg
MC$_4$	0.2683	0.2990	0.3206	0.3224	0.3303	0.3287	0.3116
SCombMNZ	0.2681	0.3056	0.3068	0.3301	0.3332	0.3471	0.3152
SCombSUM	0.2714	0.2912	0.3189	0.3265	0.3355	0.3425	0.3143
RCombSUM	0.2624	0.3029	0.3018	0.2961	0.3085	0.3042	0.2960
DCombSUM	0.2657	0.3109	0.3231	0.3331	0.3365	0.3428	0.3187
DCombMNZ	0.2782	0.3111	0.3212	0.3328	0.3366	0.3428	0.3205

Table 8.6 Experimental results for TREC 9L

	2	4	6	8	10	12	Avg
MC$_4$	0.2796	0.3096	0.3178	0.3234	0.3294	0.3341	0.3157
SCombMNZ	0.2740	0.3077	0.3430	0.3563	0.3410	0.3493	0.3286
SCombSUM	0.2699	0.3161	0.3427	0.3425	0.3406	0.3460	0.3263
RCombSUM	0.2799	0.3282	0.3196	0.3358	0.3432	0.3530	0.3266
DCombSUM	0.2728	0.3055	0.3288	0.3420	0.3468	0.3477	0.3239
DCombMNZ	0.2720	0.3176	0.3254	0.3435	0.3454	0.3477	0.3253

Table 8.7 Experimental results for TREC 2001

	2	4	6	8	10	12	Avg
MC_4	0.2652	0.2927	0.3099	0.3134	0.3169	0.3172	0.3025
SCombMNZ	0.2599	0.2884	0.3058	0.3176	0.3156	0.3231	0.3017
SCombSUM	0.2598	0.2886	0.3084	0.3172	0.3204	0.3241	0.3031
RCombSUM	0.2567	0.2884	0.2847	0.2877	0.2971	0.2994	0.2857
DCombSUM	0.2614	0.2942	0.3096	0.3184	0.3237	0.3268	0.3057
DCombMNZ	0.2637	0.2979	0.3090	0.3194	0.3228	0.3268	0.3066

Table 8.8 Averaged results over the four TREC collections

The best performing technique of each fusion is marked by a shaded background (in yellow, if this document is viewed in colour) in the corresponding table cell. When our two techniques are first and second best, they are both shaded. It can be seen that both DCombSUM and DCombMNZ, and especially the latter, are globally better than the other techniques. DCombMNZ is only surpassed on average in the TREC 8 and 9 by MC_4 and in TREC 2001 by SCombMNZ, while the performance of DCombSUM, which could be thought of as a non-tuned version of our algorithm, performs slightly below DCombMNZ, but still globally better than any other of the Renda & Straccia algorithms. In most cases where DCombMNZ or DCombSUM are not the top methods, the difference is small and they are second or consistently well positioned. This is clearly evidenced in the last table, where the averaged results over the four collections are shown.

On the other hand, SCombSUM and SCombMNZ differ from our two methods DCombSUM and DCombMNZ only in the normalization step (standard normalization vs. our method). Similarly, RCombSUM differs from DCombSUM in that it uses Rank-sim normalization. Therefore, the **most significant comparisons** are a) SCombSUM and RCombSUM against DCombSUM, and b) SCombMNZ against DCombMNZ, because they allow to evaluate the improvement introduced by our normalization technique vs. the standard one and Rank-sim, respectively, on different data sets, and under different combination methods. MC_4 cannot be compared at the normalization step, since it does not have any, but it is included here for the sake of overall comparison with the top performing technique in (Renda & Straccia, 2003).

It can be seen that when used with CombMNZ, our technique performs better than standard score normalization in 18 out of 24 trials. When used with CombSUM, our technique beats standard score normalization in 18 out of 24 trials, and Rank-sim in 20 out of 24. The values that beat our method are highlighted in boldface (in red, if this document is viewed in colour) on the tables. It can be seen that in these cases the difference is scant. As a consequence, on average, in table 8, our technique is the winner w.r.t. a) and b) in all trials.

Based on the same data, Fig 8.5 gives an idea of the size of historical data in Hs needed for the method to reach a good performance. It can be seen that the requirements are far from expensive.

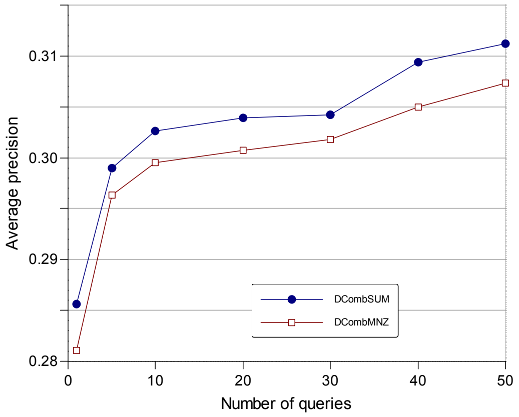


Fig 8.5 Number of runs needed to reach performance

8.5 Discussion

The work documented here deals with the general issues of automatically combining different ranked list of documents, obtained by different search methodologies as an attempt to alleviate the problem of knowledge incompleteness currently suffered by semantic retrieval engines.

Our work is motivated by the complexity and subtleties of semantic retrieval recall, which is variable considering the completeness of the semantic knowledge available for each user request, and do not play an equally important role in all situations. Solving this complexity is inherently difficult but coping with it to some degree is likely to be key for the robustness and reliability of semantic retrieval systems.

Rather than attempting to remove the difficulties of knowledge incompleteness, which would be unrealistic, the techniques proposed here aim at defining techniques that take advantage of previous keyword-based search models, so as to achieve global, relative advancements, significant enough to improve the consistency of semantic retrieval and contribute to its success as a novel technology for many users.

On another angle, our approach is novel in that it combines the implicit rankings collected at runtime. The benefit is twofold: the semantic retrieval techniques gain accuracy and reliability by each new executed query and the results obtained are filtered, enriched, and made more coherent and senseful by considering the statistical behaviour of both ontology-based and keyword-based algorithms.

The developments presented here have been tested with the available contents and the domain ontologies and metadata presented in sections 5.5 and 6.3. Besides the development and evaluation work, the possibilities for the continuation of the research are manifold. Studying score distributions is a research topic by itself. For instance, we foresee that a finer, more specialized analysis of score distributions could be achieved by identifying and separating certain conditions on which the distribution may depend, such as properties of the queries, the search space, the result set, or other domain-specific factors.

Chapter 9

Conclusions and future work

The **goal** of this thesis is the realization of a novel semantic retrieval model that exploits high-formalized semantic knowledge in the form of ontologies and KBs to improve search in large open and heterogeneous repositories of unstructured information. To achieve this goal we investigated the following **research questions**:

Q1: What do we understand by semantic search?

Q2: Where are we standing in the progress towards semantic information retrieval?

Q3: Can we combine achievements in semantic retrieval from different research fields and thereupon give rise to enhanced semantic retrieval models?

Q4: Can semantic retrieval models be scaled to open, massive, heterogeneous environments such as the World Wide Web?

Q5: How to standardize the evaluation of semantic retrieval systems?

Q6: How to deal with the problem of knowledge incompleteness?

In the first part of the thesis we study the first two questions, and establish a set of benefits and drawbacks of semantic search approaches coming from both the IR and the SW areas.

In the second part of the thesis we focus on the next three questions. With the final goal of improving the retrieval performance of traditional keyword-based search, we propose a novel semantic retrieval model that integrates and exploits formal semantic knowledge within traditional IR ranking models. Following an ambitious extension of this research line, we investigate the feasibility of applying semantic retrieval models to the Web environment. Several problems, among we can highlight the size and heterogeneity of the content or the need of simple ways of interaction with users, keep this line of research open to further improvements. To evaluate the proposed semantic retrieval system, and with the most ambitious goal of standardizing the evaluation of semantic retrieval approaches; we have constructed two different evaluation benchmark based on the Cranfield paradigm (Cleverdon, 1967) (Cleverdon, 1991). Therefore, the generated benchmarks allow the compassion not just among the semantic retrieval systems, but between semantic retrieval systems and traditional keyword-based search approaches.

In the final part of the thesis we investigate two problems arisen from the application of semantic knowledge to search environments involving a great variety of domains: heterogeneity and semantic incompleteness. The fist problem refers to the need of covering with semantic information all the domains of knowledge reflected in the contents. The second problem refers to the need of still retrieving accurate results when the semantic information is not available or incomplete.

In this chapter we describe our conclusions and contributions related to these research questions (section 9.1) and discuss our ideas for future work extensions (section 9.2).

9.1 Conclusions and contributions

Two general conclusions can be drawn from this work:

- Semantic retrieval approaches can integrate and take advantage of SW and IR views and technologies to provide better search capabilities, achieving a qualitative improvement over keyword-based retrieval by means of the introduction and exploitation of fine-grained domain ontologies.
- The application of semantic retrieval models to the Web, and more specifically the integration of ontologies as key-enablers to improve search in this environment, remains an open problem. Challenges and limitations such as the size and heterogeneity of the Web, the scarceness of the semantic knowledge, the usability constraints, or the lack of formal evaluation benchmarks, can be pointed out as some of the main reasons for the slow application of the semantic retrieval paradigm at a Web scale.

In this section we detail our major conclusions and describe our contributions to the state of the art. We organize our discussion around the six topics introduced by our research questions.

9.1.1 Semantic search: definition, classification and limitations

Chapter 4 provides a deep study of the notion of semantic search coming from both the IR and the SW areas. Towards a tentative definition, in a synthesis of the literature we point out as semantic search the idea of raising the representation of content meanings to a higher level above plain keywords, in order to enhance current mainstream Information Retrieval (IR) technologies.

Despite the large amount of work in conceptual search carried out in the IR field, semantic search has not been approached as a radically new paradigm, but as a refinement of traditional IR techniques, until the emergence of the SW. In this work, we study the strengths and weaknesses of the proposals towards the semantic search paradigm developed within both the IR and the SW fields, providing a systematic analysis of those approaches based on common features of the models and techniques under study. We propose a classification of semantic search systems from both areas attending to a set of classification criteria that includes:

- *The type of semantic information used*: linguistic conceptualization approaches, LSA approaches, and ontology-based approaches.

- *The scope of the search process*: Web search, desktop search or limited domain repositories.
- *The goal of the search process*: data retrieval, information retrieval.
- *The type of query*: keyword-based, Natural Language, semi-structured, based on ontology query languages.
- *The type of content retrieved*: pieces of ontological knowledge, XML documents, textual documents, multimedia information.
- *The type of ranking*: No ranking, keyword-based ranking, ontology-based ranking.

After this study, we analyze the main limitations of current semantic search systems and we highlight the following ones:

- *Provide a shallow representation of the information space*: they to not exploit the advantages of rich conceptualizations.
- *Reduce the information retrieval problem to a data retrieval task*: they assume that the whole information corpus can be fully represented as an ontology-driven KB.
- *Are restricted to specific domains*: they are restricted to a very concrete set of domains, and therefore, they are not applicable to heterogeneous information sources.
- *Do not deal with the problem of knowledge incompleteness*: their retrieval performance significantly drops when there is not enough available semantic information.
- *Do not exploit semantic information to improve the ranking processes*: if they return unstructured content (documents), they generally rely on traditional keyword-based retrieval models to rank them.
- *The semantic retrieval and ranking algorithms do not scale to large information sources*: the application of ontology-based retrieval on the Web remains an open problem.
- *Are impractical from the usability point of view*: they demand an excess of knowledge or feedback from users in order to express their requirements.
- *There are not standardized evaluation measures and benchmarks*: ontology-based retrieval models lack of standard evaluation benchmarks or measures to test their quality.

Our contribution is the **realization of an extensive study of the literature on semantic search approaches**, providing a classification of the different models based on their key features and identifying their main drawbacks and limitations**.**

9.1.2 Our proposal towards semantic retrieval

Chapter 5 presents our proposal towards the development of semantic retrieval models. Its main goal is to exploit highly formalized semantic knowledge in the form of ontologies and KBs to improve traditional keyword-based search over large document repositories.

Taking advantage of the years of experience and research in the IR field, the proposed semantic retrieval approach is based on an adaptation of the classic vector-space model, where keywords are replaced by semantic entities (senses).

The model includes a semantic indexing or annotation algorithm that associates the semantic entities (senses) to the documents. This can be seen as an adaptation of traditional inverted indices, where instead of keywords associated to documents we have semantic entities (senses). The annotations weights, or relevance of the semantic entities within documents, are computed using an adaptation of another traditional IR measure, the TF-IDF.

Queries are expressed using ontology-based query languages. This allows on the one hand, to articulate more expressive queries and, on the other hand, to retrieve the exact answers to the users' requests in the form of pieces of ontological knowledge (if available).

A ranking algorithm is included in this model which exploits the conceptualizations involved in queries and contents. In this ranking model, ontology-based retrieval is combined with conventional keyword-based retrieval to achieve tolerance to knowledge incompleteness.

This approach is tested on a corpus of a significant scale, showing clear improvements with respect to keyword-based search.

Our contribution is the **implementation of a novel ontology-based retrieval model,** which exploits rich semantic representations in the form of domain ontologies and KBs, supporting semantic retrieval in large repositories of unstructured information**.**

9.1.3 Advancing towards semantic retrieval in the Web environment

As pointed in chapter 5, the adaption of semantic search models to large-scale, dynamic and heterogeneous environments such as the Web introduces several research challenges:

- **Usability**: semantic search systems in general demand from the user previous knowledge on ontology-based queries, or the use of complicate form-based interfaces to express their requirements.
- **Scalability**: semantic search systems in general do not scale to massive information sources: some systems are based the ideal view where all the unstructured contents are formalized in terms of semantic knowledge, which is clearly not feasible nowadays at a Web scale. Other systems do not deal with the problem of knowledge incompleteness and their retrieval performance significantly decreases when the semantic information is not available of incomplete.
- **Heterogeneity**: semantic search systems are generally limited to the use of a predefined set of ontologies and therefore, do not cover all the potential domains involved in the Web contents.

Our contribution is based on providing potential solutions to the above mentioned problems, **taking a step towards the advancements of semantic retrieval models within large-scale and heterogeneous environments such as the Web**. This goal has been achieved by:

- **The integration of an external NL query processing module, PowerAqua** (Lopez, Motta, & Uren, 2006). This integration aims to solve the problem of usability, allowing the user to express his requirements in natural language, and the problem of heterogeneity, exploiting PowerAqua's ability to answer queries using large amounts of semantic content.
- **The implementation of flexible and scalable annotation algorithms** that generate annotations between large amounts of documents and semantic metadata, maintain both information sources decoupled.
- **The generation of a SW gateway** (see chapter 7) that aims to collect, store and provide fast and common access for applications to the available SW content. The generation of this SW gateway aims to address the heterogeneity limitation, providing fast access to large amounts of semantic content.

9.1.4 Generating semantic retrieval evaluation benchmarks

Chapter 5 and 6 introduce the problem of ontology-based retrieval systems evaluation.

In contrast to traditional IR communities, where evaluation using standardized techniques, such as those prescribed by the TREC [63]annual competitions, has been common for decades, the SW community is still a long way from defining standard evaluation benchmarks that comprise all the required information to judge the quality of the current semantic retrieval methods. Current approaches for SW technologies evaluation are based on user-centered methods. (Sure & Iosif, 2002) (McCool, Cowell, & Thurman, 2005) (Todorov & Schandl, 2008). These evaluation techniques involve users to judge the quality of SW applications under specific use cases. Therefore, they tend to be high-cost, non-scalable and difficult to repeat.

Nonetheless, we wanted to test our system systematically and as rigorously as we could. To do so, we decided to evaluate our system using models and measures traditionally used by the IR community, like the Cranfield evaluation model (Cleverdon C. , 1967) and the precision, recall evaluation metrics.

Following this research line we first generated a medium-scale evaluation benchmark comprising:

- *Document corpus*: 145,316 documents (445 MB) extracted from the CNN Web site[64].
- *Semantic information*: the KIM domain ontology and KB (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004), publicly available as part of the KIM Platform, taking a total of 71 MB in RDF text format.
- *Queries*: a set of 20 queries manually designed.
- *Judgments*: judgments for each query were manually established by human evaluators on a scale from 0 to 5.

[63] http://trec.nist.gov

[64] http://dmoz.org/News/Online_Archives/CNN.com

This evaluation benchmark can be used to test other semantic retrieval and keyword-based approaches. However, it presents two main disadvantages: a) the documents, queries and judgments are not validated and standardized by a research community and b) it size is not enough to test the retrieval algorithms at a Web-scale. Attempting to overcome these limitations we generated a new large-scale Web-based evaluation benchmark adapting the standardized TREC WT10G document collection and the queries and documents provided by the TREC 9 and TREC 2001 competitions.

- *Document corpus*: TREC WT10G (10GB of crawled Web pages).
- *Semantic information*: 40 public ontologies on the SW, potentially covering a subset of the TREC domains and queries (some of them semi-automatically populated from Wikipedia), comprising 400 MB of metadata plus 2GB of additional semantic information.
- *Queries*: 20 queries from the TREC 9 and TREC 2001 competitions.
- *Judgments*: judgments for each query where provided as part of the TREC 9 and TREC 2001 competitions.

Our contribution here is **the development of widely applicable ontology search evaluation benchmarks based on standardized IR resources and reusing online available SW data.**

9.1.5 Dealing with knowledge incompleteness

As we have previously pointed during the development of this thesis work, the semantic knowledge incompleteness is still and open problem we should face if we want to achieve successful semantic retrieval approaches.

Our proposal towards this problem relies on combining in a final ranking list, the results obtained by means of our ontology-based retrieval algorithms and a traditional keyword-based search approach. Thus, we retain keyword-based search recall when the available semantic information is scarce or incomplete.

This combination presents several interesting research issues. The ontology-based ranking algorithms behave very differently depending on whether there is sufficient semantic information to answer the user's query or not. If so, the results returned by the ontology-based approach are significantly more accurate than those obtained by the keyword-based approach. In such cases, the combination should not be uniform, but biased to the ontology-based results. The opposite situation occurs when the available semantic information is not enough to answer the user's request. In such cases, the combination should be biased to the keyword-based results. In the remaining cases, a compromise must be achieved in the combination to provide the best possible ranking list.

The target is therefore to study different techniques of ranking fusion to further enhance the reliability and robustness of the combined retrieval performance.

Our contribution here is to **tackle the problem of knowledge incompleteness towards the proposal of a novel rank fusion approach that takes into consideration statistical information in order to improve the combination of rankings coming from ontology-based and keyword-based search results.**

9.2 Discussion and future work

The general idea of introducing higher levels of explicit semantics in IR systems remains an open problem for research and discussion today, not just with regards to achieving good solutions, but also in the definition and understanding of the problem itself.

In this thesis we have studied the concept of semantic retrieval from both the IR and SW perspectives. At the outset of the research undertaken in this thesis the perception is that the works in information search and retrieval from the semantic-based technology (a.k.a. Semantic Web) area have not yet taken full advantage of the technologies, background, knowledge, and accumulated experience through several decades of work in the IR field tradition. One might say there is even some mismatch sometimes in the understanding of ground notions in both areas, such as information need, relevance, retrieval task, methodological soundness, etc. Interesting research opportunities would hence lie in the integration of perspectives from both fields in mutual benefit.

Starting from this position, we have investigated the definition of ontology-based IR models, oriented to the exploitation of domain KBs to support semantic search capabilities in document repositories, stressing on the one hand the use of ontologies and KBs in the semantic-based perspective, and on the other the consideration of unstructured content as the final search space. In other words, we have explored the use of semantic information to support more expressive queries and more accurate results, while the retrieval problem is formulated in a way that is proper of the IR field, thus drawing benefit from the state of the art in this area, and enhancing the applicability and suitability of approaches to realistic settings.

Tough we have covered a considerable number of the most important problems, further important research topics lie ahead which are not addressed here, but have a close relation to the ones addressed. This includes incremental improvements, alternative aspects to the presented proposals, as well as entire new lines of work.

Unsolved limitations, possible courses of action to address them, and potential future research challenges are discussed in the subsections that follow.

9.2.1 Semantic resources

The effectiveness of semantic retrieval systems strongly depends on the richness of the metadata representation in the ontologies and KBs, and the quality of the item annotations.

The **difficulties and cost of building and maintaining rich semantic resources** (a case of the often referred to as *knowledge acquisition bottleneck)* is a well-known fundamental hurdle, already identified by quite earlier times in the field (Croft, 1986). Even if radical effectiveness enhancements have proved to be achievable, their degree is obviously in direct relation to the amount and quality of information procured by the resource at hand (thesauri, ontologies, KBs, etc.). A fundamental issue here is to discern what expectation on the detail (depth) and coverage (breadth) would be appropriate to be realistically assumed or aimed at, and how well we may cope with the remaining incompleteness beyond that point, which can be considered a requirement to any system using such hard to procure resources. The way to satisfy the latter in our model (as described in chapters 5 and 6) is by means of a graceful degradation to a classic IR system which gets by without semantics when they are insufficient.

With respect to the quality and detail of the semantic information sources, the revised approaches adopt notably varying points of view, from LSA techniques (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990), which do completely without any source besides the document corpus itself, or Spärck Jones' "taking words as they stand" principle (Spärck Jones K. , 2003), less strict in practice, to the approaches based on **domain ontologies**, which generally tend to the opposite extreme, proposing the intensification of domain knowledge, often assuming a considerable level of detail and formalization (Maedche A. , Staab, Stojanovic, Studer, & Sure, 2003), and intermediate approaches which make do with more generic or superficial resources (Paice, 1991). The most ambitious perspectives on knowledge quality often raise controversy with respect to their feasibility, as they indeed posit hypothesis which are difficult to grant a priori. Notwithstanding, some level of automation is considered in most cases, as a palliative of the semantic resource construction cost, as well as some compromise on the scale and generality of the application scenarios (e.g. intranet-oriented scenarios, or restricted domain), with a view to their feasibility.

The design and construction of ontologies are outside the scope of the objectives of this thesis, and are subjects of extensive study in various disciplines of the SW area (Gómez-Pérez, Fernández-López, & Corcho, 2003). The semantic retrieval model proposed here started from a set of already built domain ontologies. For example, the first proposal (chapter 5) was tested with the KIM domain ontology and KB (Kiryakov, Popov, Terziev, Manov, & Ognyanoff, 2004), external to this thesis. For the extension to the Web environment (chapter 6) an own SW gateway was developed to exploit semantic content available on the Web. Many of the collected ontologies contain the definition of class hierarchies, properties and relations, but do not contain any instance. For this reason we devised a semi-automatic population method which exploits Wikipedia lists and tables (see section 6.3.1.1). As future research line we plan to explore other SW gateways, such as Watson (D'Aquin, Baldassarre, Gridinoc, Angeletou, Sabou, & Motta, 2007) to **make use of larger amounts of online available semantic metadata**.

Another important research problem, aside from obtaining high quality knowledge resources, is the annotation of unstructured content. The annotation problem consists in identifying semantic entities within the contents. It is a difficult research problem on its own, which is being widely studied in areas such as IR, NLP and SW. In this thesis, the annotation process has been addressed in different ways (sections 5.2.1, 6.2.1.1 and 6.2.1.2). All these proposals introduce a small analysis to detect ambiguity situations in which we detect the wrong meaning of the concepts. One of the strategies, explained in section 6.2.1.2, applies a strong ambiguity detection algorithm, resulting in the loss of a certain amount of relevant annotations, which, as shown in the experiments, negatively impacts the semantic retrieval performance. A **deeper study would be worthwhile to shed further light on the trade-off between the quality and quantity of annotations**.

9.2.2 Extensions

Several different interesting research lines can be studied to enhance or current semantic retrieval model. Among these extensions we can include future lines within the areas of personalization, contextualization and recommendation models.

- **Extending the model to include personalization**: personalization in information retrieval aims at improving the user's experience by incorporating the user subjectivity into the retrieval methods and models. The exploration of implicit user interest and preferences is an interesting research line to enhance our current semantic retrieval model adapting or reranking the final answers according to user-preferences.
- **Extending the model to include contextualization**: Personalized content retrieval aims at improving the retrieval process by taking into account the particular interests of individual users. However, not all user preferences are relevant in all situations. It is well known that human preferences are complex, multiple, heterogeneous, changing, even contradictory, and should be understood in context with the user goals and tasks at hand. This research extension proposes to exploit the semantics of our retrieval model to build a dynamic representation of the semantic context, information obtained from the search history and the user interaction with the system. The ontology-driven representation of the domain of discourse given by our semantic retrieval model may be seen as an important advantage to provide enriched descriptions, enabling the definition of effective means to relate preferences and context.
- **Extending the model to include recommendations**: Recommender systems suggest user products or services they may be interested in, by taking into account or predicting their tastes, priorities and goals. Commercial applications like *Amazon online store* (www.amazon.com), *Google News* (news.google.com) or *YouTube* (www.youtube.com), are several examples of successful recommender systems. An interesting future line of this work will be the research towards concept-based recommendation strategies that improve or complement the capabilities of our semantic search system, recommending contents (without query), or adapting semantic query answers, according to other user's preferences and content ratings.

Appendix A

Acronyms

The following are the acronyms used throughout this document. For each of them, a brief description of its meaning is provided. In most cases, the presented descriptions have been obtained from Wikipedia (wikipedia.org).

AI *Artificial Intelligence*, the intelligence of machines and the branch of computer science that aims to create it.

API *Application Programming Interface*, a set of declarations of the functions (or procedures) that an operating system, library or service provides to support requests made by computer programs.

DARPA *Defence Advanced Research Projects Agency*, an agency of the United States Department of Defence responsible for the development of new technology for use by the military.

HTML *HyperText Markup Language*, the predominant markup language for Web pages.

IE *Information Extraction,* the process of extracting user-specified text from a set of documents.

IPTC *International Press Telecommunications Council*, a consortium of the world's major news agencies and news industry vendors. It develops and maintains technical standards for improved news exchange.

IR *Information Retrieval*, the science of searching for information in documents, searching for documents themselves, searching for metadata that describe documents, or searching within databases.

KB *Knowledge Base*, a special kind of database for knowledge management, which provides the means for the computerized collection, organization, and retrieval of knowledge.

KR *Knowledge Representation,* the study of how knowledge about the world can be represented and what kinds of reasoning can be done with that knowledge.

LSA *Latent Semantic Analysis*, a technique in natural language processing of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.

NLP *Natural Language Processing*, a subfield of Artificial Intelligence and Computational Linguistics, which studies the problems of automated generation and understanding of natural human languages.

OWL *Web Ontology Language*, a markup language for publishing and sharing data using ontologies on the World Wide Web.

RDF *Resource Description Framework*, a World Wide Web Consortium specification for a metadata model and component in the proposed Semantic Web.

RDQL *RDF Query Language*, a computer language able to retrieve and manipulate data stored in Resource Description Framework (RDF) format.

SPARQL *SPARQL Protocol and RDF Query Language (recursive acronym)*, a RDF query language and data access protocol for the Semantic Web. On 15th January 2008, SPARQL became an official W3C Recommendation.

SQL *Structured Query Language*, a database computer language designed for the retrieval and management of data in relational database management systems, database schema creation and modification, and database object access control management.

SVD *Singular Value Decomposition*, an important factorization of a rectangular real or complex matrix, with several applications in signal processing and statistics. Applications which employ the SVD include computing the pseudo-inverse, least squares fitting of data, matrix approximation, and determining the rank, range and null space of a matrix.

SW *Semantic Web:* is an evolving extension of the World Wide Web in which the semantics of information and services on the Web is defined, making it possible for the Web to understand and satisfy the requests of people and machines to use the Web content.

VSM *Vector-space Model:* a text IR model where documents and queries are represented as vectors in a t-dimensional space

W3C The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web (WWW). It is arranged as a consortium where member organizations maintain full-time staff for the purpose of working together in the development of standards for the World Wide Web

WWW *World Wide Web:* The World Wide Web (commonly shortened to the Web) is a system of interlinked hypertext documents accessed via the Internet. With a Web browser, a user views Web pages that may contain text, images, videos, and other multimedia and navigates between them using hyperlinks.

Appendix B

Test User Interface

This appendix describes the User Interface (UI) of the implementation of the proposed approaches with which the semantic retrieval models where tested and evaluated. Two different UI are shown in this appendix. Section B.1 shows all the components of the original semantic retrieval system interface, developed in collaboration with David Vallet (chapter 5). Section B.2 displays, the PowerAqua (Lopez, Motta, & Uren, 2006) system interface. This interface is displayed to explain the changes that were made within the system (chapter 6) to address the usability limitations presented in the first version of our ontology-based retrieval model.

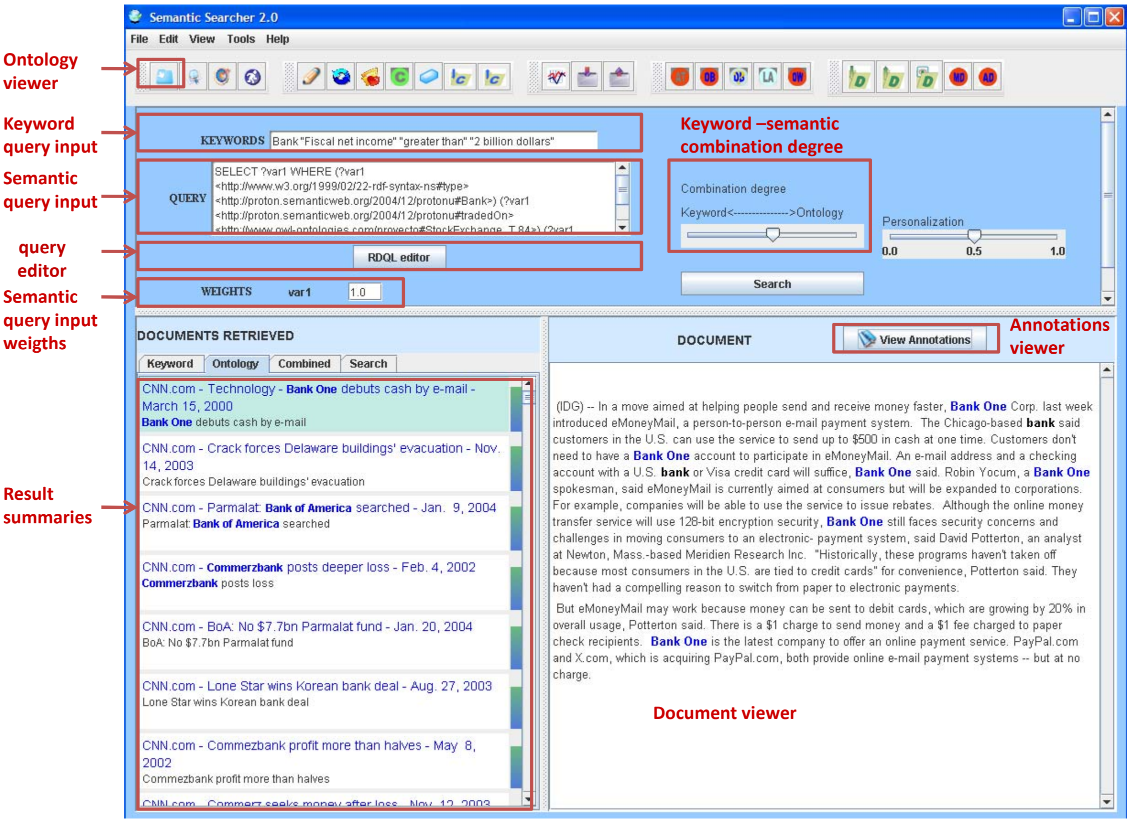
B.1 Semantic retrieval system User Interface

The following snapshot, (Fig B.1), shows the main window of the UI of our system. Among the elements that compose the UI we can highlight:

Query UI

- *Keyword query input*: The user interface is prepared to receive as input a traditional keyword-based query. However, as we have seen in the previous sections, the system automatically extracts it from the SPARQL query if no other input is provided by the user.
- *Semantic query input*: The semantic query is expressed by means of an ontology-based query language, in this case SPARQL. To facilitate the semantic query construction, an SPARQL query editor is provided (see Fig B.2 and Fig B.3).
- *SPARQL query editor*: This dialog allows the interactive edition of SPARQL queries, although, not the whole query spectrum can be generated in this way. Fig B.3 show how this dialog allows the user to select which type of concept he is looking for, to add restrictions to the properties of the searched concepts and to add restrictions on the relations to these concepts with others from the KB. Fig B.2 shows this editor before and after selecting the corresponding restrictions. Note that all the semantic queries used during the evaluation of our system have been constructed with this SPARQL editor.
- *Semantic query input weights*: In our model, the variables in the SELECT clause of the SPARQL query can be weighted. These weights indicate the relative interest of the user for each of the variables to be explicitly mentioned in the documents. The text fields to introduce those

weights are dynamically generated considering the number of variables requested in the semantic query.



KEYWORDS Bank "Fiscal net income" "greater than" "2 billion dollars"

QUERY

```
SELECT ?var1 WHERE (?var1
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://proton.semanticweb.org/2004/12/protonu#Bank>) (?var1
<http://proton.semanticweb.org/2004/12/protonu#tradedOn>
<http://www.owl-ontologies.com/proyecto#StockExchange_T 84>) (?var1
```

WEIGHTS var1 1.0

CNN.com - Technology - **Bank One** debuts cash by e-mail - March 15, 2000
Bank One debuts cash by e-mail

CNN.com - Crack forces Delaware buildings' evacuation - Nov. 14, 2003
Crack forces Delaware buildings' evacuation

CNN.com - Parmalat: **Bank of America** searched - Jan. 9, 2004
Parmalat: **Bank of America** searched

CNN.com - **Commerzbank** posts deeper loss - Feb. 4, 2002
Commerzbank posts loss

CNN.com - BoA: No $7.7bn Parmalat fund - Jan. 20, 2004
BoA: No $7.7bn Parmalat fund

CNN.com - Lone Star wins Korean bank deal - Aug. 27, 2003
Lone Star wins Korean bank deal

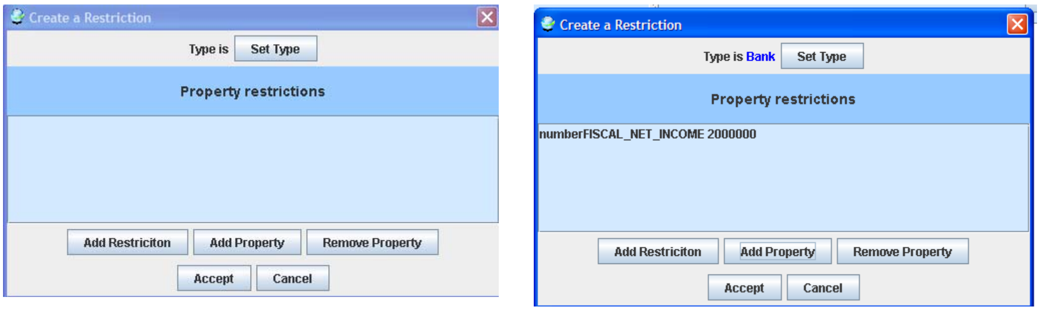
CNN.com - Commezbank profit more than halves - May 8, 2002
Commezbank profit more than halves

CNN.com - Commerz seeks money after loss - Nov. 12, 2003

(IDG) -- In a move aimed at helping people send and receive money faster, **Bank One** Corp. last week introduced eMoneyMail, a person-to-person e-mail payment system. The Chicago-based **bank** said customers in the U.S. can use the service to send up to $500 in cash at one time. Customers don't need to have a **Bank One** account to participate in eMoneyMail. An e-mail address and a checking account with a U.S. **bank** or Visa credit card will suffice, **Bank One** said. Robin Yocum, a **Bank One** spokesman, said eMoneyMail is currently aimed at consumers but will be expanded to corporations. For example, companies will be able to use the service to issue rebates. Although the online money transfer service will use 128-bit encryption security, **Bank One** still faces security concerns and challenges in moving consumers to an electronic- payment system, said David Potterton, an analyst at Newton, Mass.-based Meridien Research Inc. "Historically, these programs haven't taken off because most consumers in the U.S. are tied to credit cards" for convenience, Potterton said. They haven't had a compelling reason to switch from paper to electronic payments.

But eMoneyMail may work because money can be sent to debit cards, which are growing by 20% in overall usage, Potterton said. There is a $1 charge to send money and a $1 fee charged to paper check recipients. **Bank One** is the latest company to offer an online payment service. PayPal.com and X.com, which is acquiring PayPal.com, both provide online e-mail payment systems -- but at no charge.

Fig B.1 Semantic retrieval User Interface

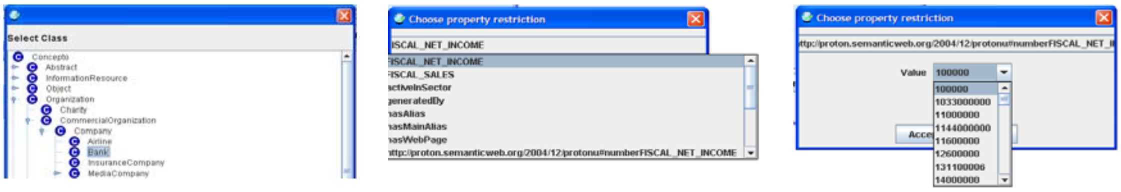


numberFISCAL_NET_INCOME 2000000

SPARQL query editor: UI for editing complex relational query conditions

SPARQL query editor: After set up one query condition: “select banks with fiscal net income greater than 2 billion dollars”

Fig B.2 SPARQL query editor: before and after setting up a query condition



Choosing type: “bank”

Choosing property: with fiscal net income

Choosing value: greater than 2 billion dollars

Fig B.3 SPARQL Editor setting up a query condition

Results display

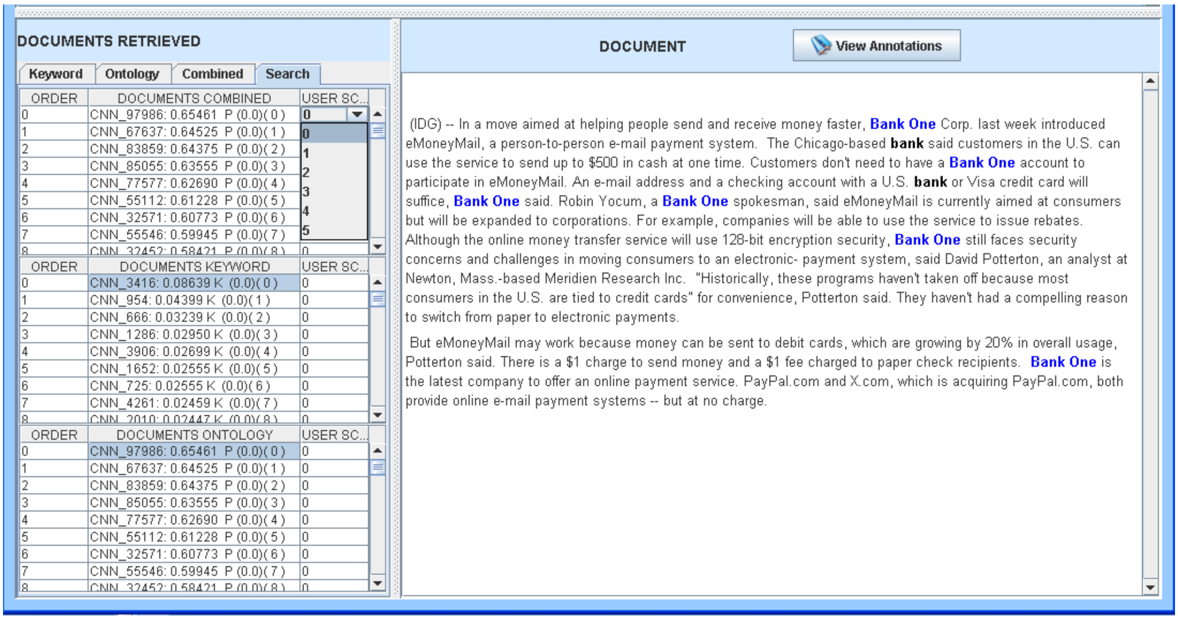
- *Result summaries*: Query results are displayed in a list of document snippets, showing the title and the date of each document. The ontological concepts involved in the documents tittles are highlighted in colour blue. As we can see in the tabs above, there are three different result summaries: the first one shows the documents retrieved by keyword-based search, the second one shows the results obtained by our ontology-only retrieval model, and the third one show she combination of both list.

- *Document viewer*: Whenever the user clicks on a document snippet, its content appears in the large text panel situated on the right part of the figure. The document is previously cleaned of HTML formatting tags. To provide a better visualization of the results in the document text, the keyword query terms are highlighted in black and the semantic query concepts are highlighted in blue.

- *Keyword-semantic combination degree slider*: As explained in section 5.2.4, to avoid the problem of knowledge incompleteness keyword-based and ontology-based results are combined in a unique ranked list. This slider allows the user to manually adjust the degree of combination showing a life reordering or results.

Evaluation UI

- *Result evaluation tab*: As we can see in Fig B.4 the evaluation tab shows the ranked list obtained for the combination of keyword-based and ontology-based retrieval, the ranked list obtained by keyword-based retrieval, and the ranked-list obtained by ontology-based retrieval. In the same row we have information about the rank position, the name of the document, the score, the personalization value (which is an extension of the model, out of the scope of this thesis) and the user rating. This rating is set by the users to evaluate the relevance of the document. The scale is establish from 0 to 5 where 0 means that the document is not relevant at all and 5 means it is highly relevant.



ORDER	DOCUMENTS COMBINED	USER SC...
0	CNN_97986: 0.65461 P (0.0)(0)	0
1	CNN_67637: 0.64525 P (0.0)(1)	0
2	CNN_83859: 0.64375 P (0.0)(2)	1
3	CNN_85055: 0.63555 P (0.0)(3)	2
4	CNN_77577: 0.62690 P (0.0)(4)	3
5	CNN_55112: 0.61228 P (0.0)(5)	4
6	CNN_32571: 0.60773 P (0.0)(6)	5
7	CNN_55546: 0.59945 P (0.0)(7)	
8	CNN_32452: 0.58421 P (0.0)(8)	0

ORDER	DOCUMENTS KEYWORD	USER SC...
0	CNN_3416: 0.08639 K (0.0)(0)	0
1	CNN_954: 0.04399 K (0.0)(1)	0
2	CNN_666: 0.03239 K (0.0)(2)	0
3	CNN_1286: 0.02950 K (0.0)(3)	0
4	CNN_3906: 0.02699 K (0.0)(4)	0
5	CNN_1652: 0.02555 K (0.0)(5)	0
6	CNN_725: 0.02555 K (0.0)(6)	0
7	CNN_4261: 0.02459 K (0.0)(7)	0
8	CNN_2010: 0.02447 K (0.0)(8)	0

ORDER	DOCUMENTS ONTOLOGY	USER SC...
0	CNN_97986: 0.65461 P (0.0)(0)	0
1	CNN_67637: 0.64525 P (0.0)(1)	0
2	CNN_83859: 0.64375 P (0.0)(2)	0
3	CNN_85055: 0.63555 P (0.0)(3)	0
4	CNN_77577: 0.62690 P (0.0)(4)	0
5	CNN_55112: 0.61228 P (0.0)(5)	0
6	CNN_32571: 0.60773 P (0.0)(6)	0
7	CNN_55546: 0.59945 P (0.0)(7)	0
8	CNN_32452: 0.58421 P (0.0)(8)	0

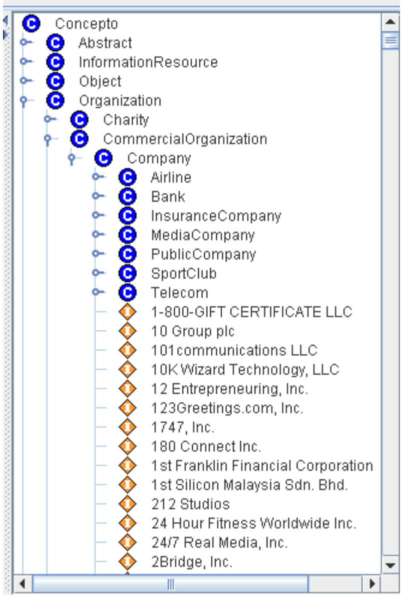
(IDG) -- In a move aimed at helping people send and receive money faster, **Bank One** Corp. last week introduced eMoneyMail, a person-to-person e-mail payment system. The Chicago-based **bank** said customers in the U.S. can use the service to send up to $500 in cash at one time. Customers don't need to have a **Bank One** account to participate in eMoneyMail. An e-mail address and a checking account with a U.S. **bank** or Visa credit card will suffice, **Bank One** said. Robin Yocum, a **Bank One** spokesman, said eMoneyMail is currently aimed at consumers but will be expanded to corporations. For example, companies will be able to use the service to issue rebates. Although the online money transfer service will use 128-bit encryption security, **Bank One** still faces security concerns and challenges in moving consumers to an electronic- payment system, said David Potterton, an analyst at Newton, Mass.-based Meridien Research Inc. "Historically, these programs haven't taken off because most consumers in the U.S. are tied to credit cards" for convenience, Potterton said. They haven't had a compelling reason to switch from paper to electronic payments.

But eMoneyMail may work because money can be sent to debit cards, which are growing by 20% in overall usage, Potterton said. There is a $1 charge to send money and a $1 fee charged to paper check recipients. **Bank One** is the latest company to offer an online payment service. PayPal.com and X.com, which is acquiring PayPal.com, both provide online e-mail payment systems -- but at no charge.

Fig B.4 Results evaluation tab

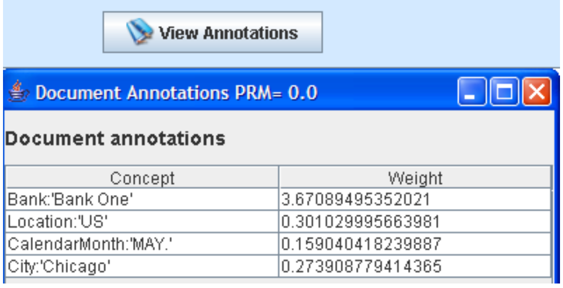
Information viewers

- *Ontology viewer:* The ontology editor allows the user to navigate across the ontology and KB, so that the user may have in mind specific details of the ontology he is using to perform the queries. As we can see in Fig B.5, concepts are represented with a blue icon and individuals are represented with an orange icon.
- *Annotations viewer:* The annotations viewer Fig B.6 shows for each document its set of annotations. For example, the document evaluated in Fig B.4 is annotated with the concepts: "bank one" instance of the class Bank, "US" instance of the class Location, "may" instance of the class CalendaryMonth and "Chicago" instance of the class City. The annotation viewer is very useful to find correlations between the quality of annotations and the relevance of the retrieved documents.



```
Concepto
  Abstract
  InformationResource
  Object
  Organization
    Charity
    CommercialOrganization
      Company
        Airline
        Bank
        InsuranceCompany
        MediaCompany
        PublicCompany
        SportClub
        Telecom
        1-800-GIFT CERTIFICATE LLC
        10 Group plc
        101communications LLC
        10K Wizard Technology, LLC
        12 Entrepreneuring, Inc.
        123Greetings.com, Inc.
        1747, Inc.
        180 Connect Inc.
        1st Franklin Financial Corporation
        1st Silicon Malaysia Sdn. Bhd.
        212 Studios
        24 Hour Fitness Worldwide Inc.
        24/7 Real Media, Inc.
        2Bridge, Inc.
```

Fig B.5 Ontology viewer



Document Annotations PRM= 0.0

Document annotations

Concept	Weight
Bank:'Bank One'	3.67089495352021
Location:'US'	0.301029995663981
CalendarMonth:'MAY.'	0.159040418239887
City:'Chicago'	0.273908779414365

Fig B.6 Annotations Viewer

B.2 PowerAqua User Interface

As we explained in chapter 6, the UI was modified to address usability issues. The following image displays the user interface of the query processing module integrated in the second version of our semantic retrieval system, PowerAqua (Lopez, Motta, & Uren, 2006). Fig B.7 shows the PowerAqua's UI where four main components can be distinguished:

Query UI

- *Natural Language query input*: The user interface is prepared to receive as input a natural language query. This new way of consultation is a compromise between usability (there's no need for the user to have previous knowledge about ontology-based query languages, or to navigate across different interfaces to express his requirements) and expressivity (the user can still express relations and properties using natural language).

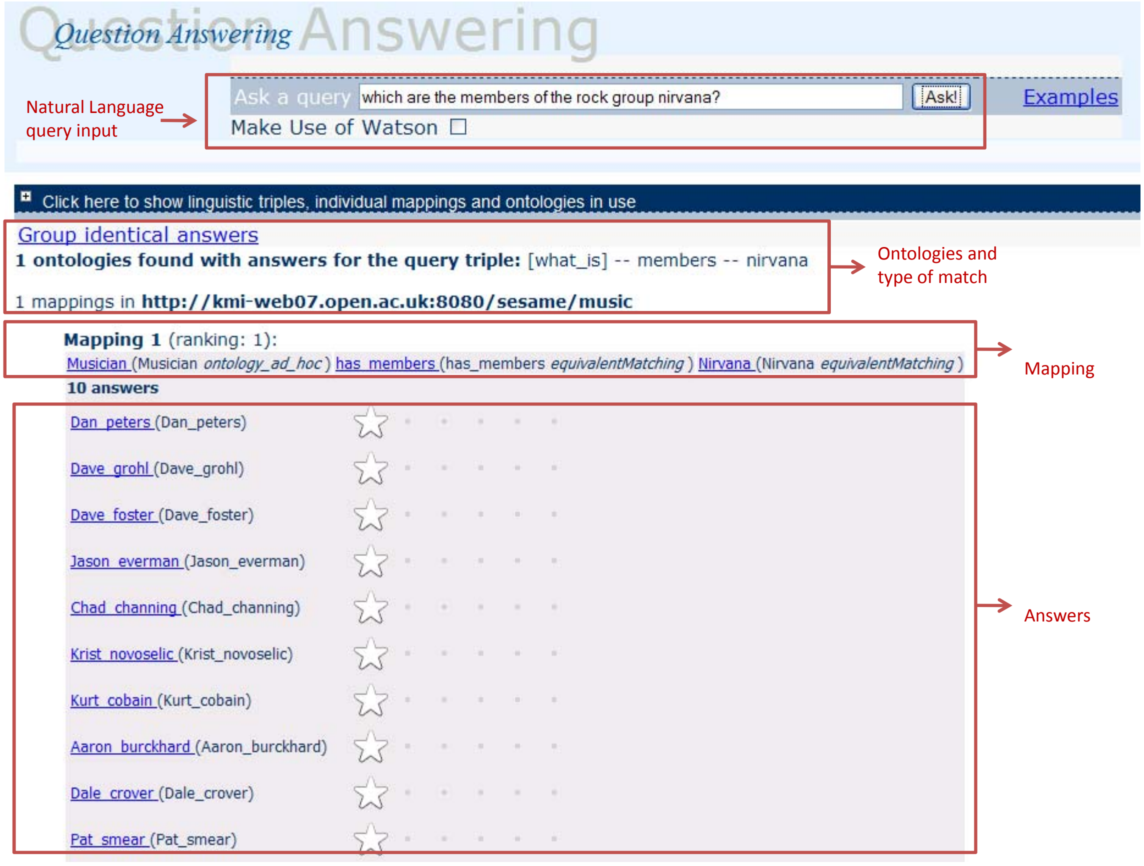
Results display

- *Ontologies and types of match*: This part of the user interface displays the syntactic triple that has matched and in which specific ontology of all the available semantic metadata.

- *Mapping*: The mapping section shows the translation of the matching in the previous selected ontology.

- *Answer*: The answer section displays the individuals or concepts found in the ontology that answer the triple extracted from the original user query.

This new interface basically replace our original query user interface elements, and adds an additional section to display the pieces of ontological knowledge extracted by PowerAqua as answer to the user's query.



which are the members of the rock group nirvana?

Make Use of Watson □

⊞ Click here to show linguistic triples, individual mappings and ontologies in use

Group identical answers

1 ontologies found with answers for the query triple: [what_is] -- members -- nirvana

1 mappings in **http://kmi-web07.open.ac.uk:8080/sesame/music**

Mapping 1 (ranking: 1):

Musician (Musician *ontology_ad_hoc*) has_members (has_members *equivalentMatching*) Nirvana (Nirvana *equivalentMatching*)

10 answers

Dan_peters (Dan_peters)

Dave_grohl (Dave_grohl)

Dave_foster (Dave_foster)

Jason_everman (Jason_everman)

Chad_channing (Chad_channing)

Krist_novoselic (Krist_novoselic)

Kurt_cobain (Kurt_cobain)

Aaron_burckhard (Aaron_burckhard)

Dale_crover (Dale_crover)

Pat_smear (Pat_smear)

Natural Language query input

Ontologies and type of match

Mapping

Answers

Fig B.7 PowerAqua's User Interface

Appendix C

Adaptation of TREC queries

The following appendix shows the adaptation of TREC queries performed in the experiments carried out in chapter 6 to adapt the standard IR evaluation topics to natural language queries accepted by PowerAqua (Lopez, Motta, & Uren, 2006). Here we list the selected TREC topics, each topic has associated a set of basic NL questions (obtained from the title, description and narrative provided by TREC) and ontologies that cover the answer. All the ontologies can be found online under http://kmi-web07.open.ac.uk:8080/sesame.

Number: 452
Question: Do beavers live in salt water?
Description: Describe the normal habitat for beavers; note exceptions, if any.
Narrative: Relevant documents describe the habitat range as well as references to specific areas and bodies of water.
Translation: Describe the habitat for beavers.
ontologies (domain) : tapfull (animals)

Number: 454
Question: Parkinson's disease
Description: What are the symptoms and treatment of Parkinson's Disease, and what segments of the population have this disease?
Narrative: Documents discussing research projects and funding for research projects were considered relevant only when clinical trials were included. Documents regarding legislation which discussed funding and programs were considered irrelevant.
Translation: What are the symptoms of Parkinson? / What is the treatment for Parkinson?
ontologies (domain) :tapfull (diseases)

Number: 457
Question: Chevrolet trucks
Description: Find documents that address the types of Chevrolet trucks available.
Narrative: Relevant documents must contain information such as: the length, weight, cargo size, wheelbase, horsepower, cost, etc.
Translation: Find chevrolets

ontologies (domain) :tapfull, autos (autos)

Number: 465
Question: Deer
Description: What kinds of diseases can infect humans due to contact with deer or consumption of deer meat?
Narrative: Documents explaining the transference of Lyme disease to humans from deer ticks are relevant.
Translation: What deer diseases can infect humans? / What human diseases are transferred by deers?
ontologies (domain) : tapfull (diseases)

Number: 467
Question: dachshund dachshunds "wiener dog"
Description: Identify documents that contain information on buying and owning dachshund dogs.
Narrative: Documents that discuss general dog information which is directly applicable to buying and owning dachshunds (i.e., how to chose a breeder) are relevant. Documents that list names of dachshund breeders and names of clubs for dachshund owners are relevant.
Translation: Show me all information about dachshund dog breeders
ontologies (domain) :danchundogs, tapfull (animals)

Number: 476
Question: Jennifer Aniston
Description: Find documents that identify movies and/or television programs that Jennifer Aniston has appeared in.
Narrative: Relevant documents include movies and/or television programs that Jennifer Aniston has appeared in.
Questions: Show me the movies of Jenifer Aniston
ontologies (domain) : movie_database (cinema)

Number: 484
Question: auto skoda
Description: Skoda is a heavy industrial complex in Czechoslovakia. Does it manufacture vehicles?
Narrative: Relevant documents would include references to historic and contemporary automobile and truck production. Non-relevant documents would pertain to armament production.
Translation: Show me the auto production of Skodas
ontologies (domain) :auto (AUTOS)

Number: 489
Question: calcium
Description: How do members of the medical profession view the effectiveness of calcium supplements?
Narrative: Any document which cites the benefits of humans using calcium supplements or advises how calcium supplements should be used are relevant. A relevant document must establish that the information comes from a qualified medical source and not from the claims of a manufacturer or vendor of calcium supplements or from the opinion of anyone not recognized by the medical profession.
Translation: What is the effectiveness of calcium supplements? / What are the benefits of calcium?
ontologies (domain) :fungalv2 (MEDICINE)

Number: 491
Question: Japanese Wave
Description: Identify occurrences in which a Japanese wave or tsunami caused loss of life or damage.
Narrative: Any reports that describe the occurrence of a Japanese wave or tsunami causing loss of life or damage are relevant. A relevant report must describe an actual event occurring at any location.
Translation: Show me all tsunamis
ontologies (domain) : phenomenon (NATURAL DISASTERS)

Number: 494
Question: nirvana
Description: Find information on members of the rock group Nirvana.
Narrative: Descriptions of members' behavior at various concerts and their performing style is relevant. Information on who wrote certain songs or a band member's role in producing a song is relevant. Biographical information on members is also relevant.
Translation: Show me all members of the rock group nirvana / What are the members of nirvana?
ontologies (domain) : tapfull, music (MUSIC)

Number: 504
Question: information about what manatees eat
Description: Find documents that describe the diet of the manatee.
Narrative: Relevant documents will identify any foods providing sustenance to the manatees.
Translation: What is the diet of the manatee? (no answer)
ontologies (domain) :tap (animals)

Number: 508
Question: hair loss is a symptom of what diseases
Description: Find diseases for which hair loss is a symptom.
Narrative: A document is relevant if it positively connects the loss of head hair in humans with a specific disease. In this context, "thinning hair" and "hair loss" are synonymous. Loss of body and/or facial hair is irrelevant, as is hair loss caused by drug therapy.
Translation: What diseases have symptoms of hair loss?
ontologies (domain) : biomedical(medicine)

Number: 511
Question: diseases caused by smoking?
Description: What diseases does smoking cause?
Narrative: A relevant document must describe smoking tobacco products as a cause of a disease. Diseases caused by second-hand smoke and smokeless tobacco are not relevant.
Translation: What diseases does smoking cause? / What diseases are caused by smoking?
ontologies (domain) : biomedical (medicine)

Number: 512
Question: how are tornadoes formed?
Description: How are tornadoes formed?
Narrative: A relevant document will provide the meteorological and atmospheric conditions necessary to create a tornado and explain how the conditions interact to form the funnel-shaped cloud.
Translation: how are tornadoes formed / Describe the formation of tornadoes
ontologies (domain) : phenomenon (natural disasters)

Number: 513
Question: earthquakes?
Description: What causes earthquakes, and where do they occur most often?
Narrative: A relevant document will discuss scientific causes of earthquakes or tremors and/or report geographic areas where earthquake activity occurs most frequently.
Translation: what causes earthquakes? / where do earthquakes occur?
ontologies (domain) : phenomenon (natural disasters)

Number: 516
Question: halloween?
Description: When, where, and how did Halloween evolve?
Narrative: A relevant document will discuss the origin of Halloween and the original customs of Halloween. Modern day trick-or-treating stories are not relevant.
Translation: What is the origin of halloween? / What are the original customs of halloween?
ontologies (domain) :stconcepts (festivities)

Number: 519
Question: info on where frogs live
Description: Find documents that describe the habitat of frogs.
Narrative: A relevant document will identify the natural habitat of any type of frog. A frog's diet is not relevant.
Translation: Where do frogs live? / Describe the habitats for frogs?
ontologies (domain) :animals- (animals)

Number: 523
Question: facts about the five main clouds?
Description: How are the five main types of clouds formed?
Narrative: A document that explains the process of cloud formation for any of the five main types of clouds is relevant. A document that discusses clouds, but does not explain their formation processes is not relevant.
Translation: How are the clouds formed? / Describe the formation of clouds**.**
ontologies (domain) :phenomenon (natural world)

Number: 524
Question: how to erase scar?
Description: What methods are used for removal of scar tissue?
Narrative: A relevant document must disclose the name of a procedure or describe it, or identify the instrument used to remove scar tissue or skin defects. Mere references to "surgical removal" are insufficient.
Translation: How to erase a scar?/ How to remove a scar?
ontologies (domain) : galen (medicine)

Number: 526
Question: bmi
Description: What does BMI stand for?
Narrative: Any document that gives defines or explains BMI is relevant.
Translation: what is BMI?
ontologies (domain) : form_demo (medicine)

References

Agosti, M., Crestani, F., Gradenigo, G., & Mattiello, P. (1990). An approach to conceptual modelling of IR auxiliary data. *IEEE International Conference on Computer and Communications.* Scottsdale, Arizona.

Agosti, M., Melucci, M., & Crestani, F. (1995). Automatic authoring and construction of hypertext for Information Retrieval. *ACM Multimedia Systems* , 15-24.

Aguirre, E., Ansa, O., Hovy, E., & Martínez, D. (2000). Enriching very large ontologies using the WWW. *First Workshop on Ontology Learning OL-2000. 14th European Conference on Artifical Intelligence. ECAI-2000.* Berlin: Germany.

Alfonseca, E., Moreno-Sandoval, A., Guirao, J. M., & Ruiz-Casado, M. (2006). The Wraetlic NLP Suite. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006).* Genoa, Italy.

American National Standards Institute, I. (1980). *Guidelines for Thesaurus Structure, Construction, and Use.* New York: ANSI Z39.19-1980.

Anderson, C. (2006). The Long Tail. Why the Future of Bussines is Selling Less of More. *Hyperion* .

Aslam, J. A., & Montague, M. (2001). Models for metasearch. *24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, (pp. 276-284). New Orleans, Louisiana, USA.

Aslandogan, Y. A., & Yu, C. T. (2000). Multiple evidence combination in image retrieval: Diogenes searches for people on the Web. *23rd Annual ACM Conference on Research and Development in Information Retrieval (SIGIR 2000)*, (pp. 88-95). Athens, Greece.

Baeza Yates, R., & Ribeiro Neto, B. (1999). *Modern Information Retrieval.* Harlow, UK: Addison-Wesley.

Bailey, P., Craswell, N., & Hawking, D. (2003). Engineering a multi-purpose test collection for web. *Information Processing and Managment* , 853-871.

Bartell, B. T. (1994). *Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval.* PhD thesis, University of California, San Diego.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American* .

Bernstein, A., & Kaufmann, E. (2006). Gino - a guided input natural language ontology editor. *5th International Semantic Web Conference.* Athens, GA, USA: Springer Verlag.

Berretti, S., Del Bimbo, A., & Pala, P. (2004). Merging Results for Distributed Content Based Image Retrieval. *Multimedia Tools and Applications* , 215-232.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using Linear Algebra for Intelligent Information Retrieval. *IAM Review archive , 27* (4), 573-595.

Bisson, G., Nédellec, C., & Cañamero, D. (2000). Designing Clustering Methods for Ontology Building. *First Workshop on Ontology Learning OL-2000. The 14th European Conference on Artificial Intelligence. ECAI 2000.* Berlin, Germany.

Bizer, C. (2007). Querying Wikipedia like a Database. *Developers track presentation. 16th International World Wide Web Conference. WWW2007.* Banff, Alberta, Canada.

Brank, J., Grobelnik, M., & Mladenić, D. (2005). A survey of ontology evaluation techniques. *SIKDD at multiconference Information Society (IS 2005).* Ljubljana, Slovenia.

Breitman, K. K., Casanova, M. A., & Truszkowski, W. (2007). *Semantic Web: Concepts, Technologies and Applications.* London, UK: Springer-Verlag.

Brewester, C. (2004). Data driven ontology evaluation. *International Conference on Language Resources and Evaluation (LREC 2004).* Lisbon, Portugal.

Buckley, C., & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. *27th annual international ACM SIGIR conference on Research and development in information retrieval*, (págs. 25 - 32). Sheffield, United Kingdom.

Burger, J. (2001). *Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A).* DARPA/NSF committee publication.

Cantador, I., Fernández, M., & Castells, P. (2007). Improving Ontology Recommendation and Reuse in WebCORE by Collaborative Assessments. *Workshop on Social and Collaborative Construction of Structured Knowledge at the 16th International World Wide Web Conference (WWW 2007).* Banff, Canada.

Cardoso, J. (2007). The Semantic Web Vision: Where are We? *IEEE Intelligent Systems* , 22-26.

Castells. (2003). La Web Semántica. *Sistemas Interactivos y Colaborativos en la Web* , 195-212.

Castells, P. F. (2004). Neptuno: Semantic Web Technologies for a Digital Newspaper Archive. *1st European Semantic Web Symposium (ESWS 2004). 3053*, pp. 445-458. Berlin Heidelberg: Springer Verlag.

Castells, P., Corella, M. A., Vallet, D., Avrithis, Y., Mylonas, P., Izquierdo, M., et al. (2005). *Personalisation module.* aceMedia deliverable D6.6.

Castells, P., Fernández, M., & Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-based Information Retrieval. *IEEE Transactions on Knowledge and Data Engineering, Special Issue on "Knowledge and Data Engineering in the Semantic Web Era" , 19* (2), 261-272.

Castells, P., Foncillas, B., Lara, R., Rico, M., & Alonso, J. L. (2004). Semantic Web Technologies for Economic and Financial Information Management. *1st European Semantic Web Symposium (ESWS 2004)* (pp. 473-487). Berlin Heidelberg: Springer Verlag.

Chapman, R. L. (1977). *Roget's International Thesaurus.* New York: Harper and Row.

Chen, H., & Lynch, K. J. (1992). Automatic construction of networks of concepts characterizing document databases. *IEEE Trans. on Systems, Man and Cybernetics 22(5)* , 885-902.

Chirita, P. A., Gavriloaie, R., Ghita, S., Nejdl, W., & Paiu, R. (2005). Activity based metadata for semantic desktop search. *2nd European Semantic Web Conference.* Heraklion, Greece.

Christophides, V., Karvounarakis, G., Plexousakis, D., & Tourtounis, S. (2003). Optimizing taxonomic semantic Web queries using labelling schemes. *Journal of Web Semantics 1, Issue 2* , 207-228.

Cimiano, P. (2006). *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications.* New York, USA: Springer-Verlag.

Cimiano, P., Haase, P., & Heizmann, J. (2007). Porting Natural Language Interfaces between Domains -- An Experimental User Study with the ORAKEL System. *International Conference on Intelligent User Interfaces.*

Cimiano, P., Pivk, A., Schimidt-Thieme, L., & Staab, S. (2004). Learning taxonomic relations from heterogeneuos evidence. *4th Workshop on Ontology Learning and Population. 18 European Conference on Artifical Intelligence. ECAI 2004.* Valencia, Spain.

Cleverdon, C. (1967). The Cranfield tests on index language devices. *Aslib Proceedings* , 173-192.

Cleverdon, C. (1991). The significance of the Cranfield tests on index languages. *14th Annual International ACM SIGIR Conference on Research and Developement in Information Retrieval.*, (pp. 3-12). Chicago, Illinois, USA.

Cohen, P., & Kjeldsen, R. (1987). Information Retrieval by constrained spreading activation on Sematic Networks. *Information Processing & Management* , 255-268.

Cohen, S., Mamou, J., Kanza, Y., & Sagiv, Y. (2003). XSEarch: A Semantic Search Engine for XML. *29th International Conference on Very Large Data Bases*, (pp. 45-56). Berlin, Germany.

Contreras, J., Benjamins, V. R., Blázquez, M., Losada, S., Salla, R., Sevilla, J., et al. (2004). A Semantic Portal for the International Affairs Sector. *14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004). 3257*, pp. 203-215. Berlin Heidelberg: Springer Verlag.

Crescenzi, V., & Mecca, G. (2004). Automatic information extraction from large websites. *Journal of the ACM (JACM)* , 731 - 779.

Crestani, F. (1997). Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review 11(6)* , 453-482.

Cristani, M., & Cuel, R. (2005). A Survey on Ontology Creation Methodologies. *International Journal on Semantic Web and Information Systems 1, Issue 2* , 49-69.

Croft. (2000). Combining approaches to information retrieval. In *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval* (pp. 1-36). Kluwer Academic Publishers.

Croft. (1986). User-specified domain knowledge for document retrieval. *9th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR 1986)*, (pp. 201-206). Pisa, Italy.

Croft, W. B., & Harper, D. J. (1993). Knowledge-based and statistical approaches to text retrieval. *IEEE Expert: Intelligent Systems and their Applications* , 8(2):8-12.

Crouch, C. J. (1990). An approach to the Automatic Construction of Global Thesauri. *Information Processing and Management 26(5)* , 629-640.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).* Philadelphia, USA.

Daconta, M. C., Obrst, L. J., & Smith, K. T. (2003). *The Semantic Web: A guide to the future of XML, Web Services and Knowledge Management.* New York, USA: Wiley.

D'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., & Motta, E. (2007). Watson: A Gateway for Next Generation Semantic Web Applications. *6th International Semantic Web Conference (ISWC2007).* Busan, Korea.

D'Aquin, M., Gridinoc, L., Sabou, M., Angeletou, S., & Motta, E. (2007). Characterizing Knowledge on the Semantic Web with Watson. *5th International EON Workshop at International Semantic Web Conference (ISWC'07).* Busan, Korea.

D'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., et al. (2008). Towards a New Generation of Semantic Web Applications. *IEEEIntelligent Systems* , 23(3):20.

Dasiopoulou, S. (2005). *Early semantic reasoning algorithms.* aceMedia project deliverable D4.4.

Davies, J., Weeks, R., & Krohn, U. (2002). Quizrdf: search technology for the semantic Web. *workshop on RDF and Semantic Web Applications 11th International WWW Conference.* Honolulu, Hawaii, USA.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science , 41* (6), 391-407.

Deshpande, M., & Karypis, G. (2004). Item-based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems , 22* (1), 143-177.

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., et al. (2003). A Case for Automated Large Scale Semantic Annotation. (Elsevier, Ed.) *Journal of Web Semantics 1, Issue 1* , 115-132.

Ding, L., Finin, T., Joshi, A., Pan, R., & Cost, S. (2004). Swoogle: A Search and Metadata Engine for the Semantic Web. *13th Conference on Information and Knowledge Management (CIKM 2004)*, (pp. 625-659). Washington, DC, USA.

Dumais, S. (1990). *Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval. TM-ARH-017527.* Bellcore.

Dumais, S. (1994). Latent semantic indexing (LSI) and TREC-2. *2nd Text Retrieval Conference (TREC2)*, (pp. 105-116).

Dwork, D., Kumar, R., Noar, M., & Sivakumar, D. (2001). Rank aggregation methods for the web. *10th International Conference on the World Wide Web (WWW 2001)*, (pp. 613-622). Hong Kong, China.

Ellis, D. (1996). The dilemma of measurement in information retrieval research. *Journal of the American Society for Information Science* , 23-36.

Feigenbaum, E. A. (1984). Knowledge engineering: the applied side of artificial intelligence. *Symposium on Computer culture: the scientific, intellectual, and social impact of the computer*, (págs. 91-107). New York, USA.

Feigenbaum, E. A. (2003). Some challenges and grand challenges for computational intelligence. *Journal of the ACM 50(1)* , 32-40.

Feigenbaum, E. A. (1997). The art of artificial intelligence: Themes and case studies knowledge engineering. *International Joint Conference on Artificial Intelligence*, (págs. 1014-1029). Nagoya, Japan.

Fellbaum, C. (1998). *WordNet, An Electronic Lexical Database.* MIT Press.

Fernández, M., Cantador, I., & Castells, P. (2006). CORE: A Tool for Collaborative Ontology Reuse and Evaluation. *4th International Workshop on Evaluation of Ontologies for the Web (EON 2006) at the 15th International World Wide Web Conference (WWW 2006).* Edinburgh, UK.

Finin, T., Mayfield, J., Fink, C., Joshi, A., & Cost, R. S. (2005). Information retrieval and the semantic Web. *38th Annual Hawaii international Conference on System Sciences (Hicss'05)*, *4*.

Fox, E. A., & Shaw, J. A. (1993). Combination of multiple searches. *2nd Text REtrieval Conference (TREC 2)*, (pp. 243-249). Gaithersburg, Maryland, USA.

Fox, E. A., Koushik, M. P., Shaw, J., Modlin, R., & Rao, D. (1992). Combining evidence from multiple searches. *1st Text REtrieval Conference (TREC 1)*, (pp. 319-328). Gaithersburg, Maryland,USA.

Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2005). A Theoretical Framework for Ontology Evaluation and Validation. *2nd Italian Semantic Web Workshop Semantic Web Applications and Perspectives (SWAP2005).* Trento, Italy.

Gauch, S., Chaffee, J., & Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems 1, Issue 3-4* , 219-234.

Gómez-Pérez, A. (1995). Some Ideas and Examples to Evaluate Ontologies. *11th Conference on Artificial Intelligence*, (pp. 299-305). Angeles, California, USA.

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2003). *Ontological Engineering.* London, UK: Springer-Verlag.

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarrán, J. (1998). Indexing with WordNet synsets can improve Text Retrieval. *COLING/ACL Workshop on Usage of WordNet for Natural Language Processing.* Montreal, Canada.

Gracia, J., Lopez, V., D'Aquin, M., Sabou, M., Motta, E., & Mena, E. (2007). Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching. *Ontology Matching Workshop at 6th International Semantic Web Conference (ISWC 2007).* Busan, Korea.

Grobelnik, M., & Mladenic, D. (2004). Visualization of News Articles. *Informatica Journal* , 28-32.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* , 199-220.

Gruber, T. R. (2008). Collective Knowledge Systems: Where the Social Web meets the Semantic Web. *Journal of Web Semantics* .

Guarino, N. (1998). Formal Ontology and Information Systems. *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS 1998)*, (pp. 3-15). Trento, Italy.

Guarino, N., & Welty, C. (2002). Evaluating Ontological Decisions with OntoClean. *Communications of the ACM* , 61-65.

Guarino, N., Masolo, C., & Vetere, G. (1999). OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems 14, Issue 3* , 70-80.

Guha, R. V., McCool, R., & Miller, E. (2003). Semantic search. *12th International World Wide Web Conference (WWW 2003)*, (pp. 700-709). Budapest, Hungary.

Handschuh, S., Staab, S., & Ciravegna, F. (2002). S-cream – Semi-automatic Creation of Metadata. *13th International Conference on Knowledge Engineering and Knowledge Management – Ontologies and the Semantic Web (EKAW 2002). 2473*, pp. 358-372. Berlin Heidelberg: Springer Verlag.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., et al. (2000). Falcon - Boosting Knowledge for Answer Engines. *9th Text Retrieval Conference (Trec-9).*

Harbourt, A. M., Syed, E. J., Hole, W. T., & Kingsland, L. C. (1993). The ranking algorithm of the Coach browser for the UMLS Metathesaurus. *17th Annual Symposium on Computer Applications in Medical Care*, (pp. 720-724). Washington, D. C., NY.

Hawking, D. (2000). Overview of the TREC-9 Web Track. In *SIRO Mathematical and Information Sciences* (p. 87).

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. *14th International Conference on Computational Linguistics*, (págs. 539--545). Nantes, France.

Hendler, J. A. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems , 16* (2), 30-37.

Hersh, W. R., & Greenes, R. A. (1990). SAPHIRE – An information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval, and hierarchical relationships. *Computers and Biomedical Research* , 410-425.

Hersh, W. R., Hickam, D. D., & Leone, T. J. (1992). Words, concepts, or both: Optimal indexing units for automated information retrieval. *16th Annual Symposium on Computer Applications in Medical Care*, (pp. 644-648). Baltimore, MD.

Heyer, G., Laüter, M., Quasthoff, U., Wittig, T., & Wolff, C. (2001). Learning Relations using Collocations. *Workshop on Ontology Learning. 17th Internacional Join Conference on Artificial Intelligence. IJCAI 2001.* Seattle, Washington, USA.

Hovy, E. H., Gerber, L., Hermjakob, U., Junk, M., & Lin, C. Y. (2000). Question Answering in Webclopedia. *TREC-9 Conference.*

Järvelin, K., Kekäläinen, J., & Niemi, T. (2001). ExpansionTool: Concept-based query expansion and construction. *Springer Netherlands* , 231-255.

Jones, S. A. (1993). thesaurus data model for an intelligent retrieval system. *Journal of Information Science 19* , 167-178.

Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., & Scholl, M. (2002). RQL: A Declarative Query Language for RDF. *Proc. of the 11th International World Wide Web Conference (WWW 2002).* Honolulu, Hawaii, USA.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic Annotation, Indexing, and Retrieval. *Journal of Web Semantics 2, Issue 1* , 49-79.

Koll, M. (1979). WEIRD: an approach to concept-based information retrieval. *ACM SIGIR Forum* , 32 - 50.

Korfhage, R. R. (1997). *Information Storage and Retrieval.* New York, USA: John Wiley & Sons, Inc.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* , 211-240.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes* , 259-284.

Lay, J. A., & Ling, G. (2006). Semantic retrieval of multimedia by concept languages: treating semantic concepts like words. *Signal Processing Magazine, IEEE. Mar 2006. ISSN: 1053-5888 , 23*, 115-123.

Lee, J. H. (97). Analysis of multiple evidence combination. *20th ACM International Conference on Research and Development in Information Retrieval (SIGIR 97)*, (pp. 267-276). New York.

Lee, W. S. (2001). Collaborative Learning for Recommender Systems. *Proceedings of the 18th International Conference on Machine Learning*, (pp. 314-321). Williamstown, MA, USA.

Lehti, P., & Fankhauser, P. (2005). SWQL – A Query Language for Data Integration Based on OWL. *OTM Workshops. 3762*, pp. 926-935. Berlin Heidelberg: Springer Verlag.

Letsche, T. A., & Berry, M. W. (1997). Large-Scale Information Retrieval with Latent Semantic Indexing. *Information Sciences - Applications 100 Issue 1-4* , 105-137.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics - Doklady , 10*, 707-710.

Lewis, D. D., & Gale, W. A. (1994). A Sequential Algorithm for Training Text Classifiers. *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, (pp. 3-12). Dublin, Ireland.

Lopez, V., Motta, E., & Uren, V. (2006). PowerAqua: Fishing the Semantic Web. *European Semantic Web Conference.* Montenegro.

Lopez, V., Pasin, M., & Motta, E. (2005). AquaLog: An Ontology-portable Question Answering System for the Semantic Web. *European Semantic Web Conference*, (pp. 546-562). Creete, Greece.

Lopez, V., Sabou, M., & Motta, E. (2006). PowerMap: Mapping the Real Semantic Web on the Fly. *5th International Semantic Web Conference (ISWC2006).* Georgia, Atlanta, USA.

Lozano-Tello, A., & Gómez-Pérez, A. (2004). Ontometric: A method to choose the appropriate ontology. *Journal of Database Management* .

Luke, S., Spector, L., & Rager, D. (1996). Ontology-Based Knowledge Discovery on the World-Wide Web. *Internet-Based Information Systems: Papers from the AAAI Workshop. AAAI*, (pp. 96-102). Menlo Park, California.

Lux, M., Klieber, W., & Granitzer, M. (2004). Caliph & Emir: Semantics in Multimedia Retrieval and Annotation. *19th International CODATA Conference.* Berlin, Germany: The Information Society: New Horizons for Science.

Lyman, P., & Varian, H. R. (2003). *HOW MUCH INFORMATION? 2003.* California, USA: http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/.

Madala, R., Takenobu, T., & Hozumi, T. (1999). Complementing WordNet with Rogert's and Corpus-based Thesauri for Information Retrieval. *9th Conference of the European Chapter of the Association for Computational Linguistics EACL 1999*, (pp. 94-101). Bergen, Norway.

Madala, R., Takenobu, T., & Hozumi, T. (1998). The use of WordNet in information Retrieval. *Use of WordNet in Natural Language Processing Systems*, (pp. 31-37). Montreal.

Maedche, A., & Staab, S. (2000). Discovering Conceptual Relations from text. *European Conference on Artificial intelligence. ECAI 2000*, (pp. 321-325). Berlin, Germany.

Maedche, A., & Staab, S. (2002). Measuring similarity between ontologies. *ACM Conference on Information and Knowledge Management . CIKM 2002.* Virginia, USA.

Maedche, A., Staab, S., Stojanovic, N., Studer, R., & Sure, Y. (2003). SEmantic portAL: The SEAL Approach. *Spinning the Semantic Web. MIT Press* , 317-359.

Magaranaki, A., Karvounarakis, G., Christophides, V., Plexousakis, D., & Anh, T. (2002). *Ontology storage and querying.* Foundation for Research and Technology Hellas. Institute for Computer Science, Information Systems Lab.

Manmatha, R., & Sever, H. (2002). A Formal Approach to Score Normalization for Metasearch. *Human Language Technology Conference (HLT 2002)*, (pp. 88-93). San Diego, California, USA.

Manmatha, R., Rath, R., & Feng, F. (2001). Modelling score distributions for combining the outputs of search engines. *24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, (pp. 267-275). New Orleans, Louisiana, USA.

Masthoff, J. (2004). Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. *User Modeling and User-Adapted Interaction , 14* (1), 37-85.

Mayfield, J., & Finin, T. (2003). Information retrieval on the Semantic Web: Integrating inference and retrieval. *Workshop on the Semantic Web at the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval.* Toronto, Canada.

McCool, R., Cowell, A. J., & Thurman, D. A. (2005). End-User Evaluations of Semantic Web Technologies. *Workshop on End User Semantic Web Interaction. ISWC 2005.* Galway, Ireland.

Miller, A., Leacock, C., Tengi, R., & Bunker, R. T. (1993). A semantic concordance. *ARPA workshop on Human Language Technology.*

Miller, G. (1995). WordNet: A lexical database. *Communications of the ACM , 38, 11*, 39-41.

Miller, G. (1990). WordNet: An online-lexical database. *International journal of lexicography* .

Mitra, P., Kersten, M., & Wiederhold, G. (2000). A Graph-Oriented Model for Articulation of Ontology Interdependencies. *Conference on Extending Database Technology (EDBT'2000).* Konstanz, Germany.

Mizzaro, S. (1998). How many relevances in information retrieval? *Interacting With Computers* , 305-322.

Mizzaro, S. (1997). Relevance: The whole history. *Journal of the American Society for Information Science* , 810-832.

Moldovan, D. I., & Mihalcea, R. (2000). Using WordNet and Lexical Operators to Improve Internet Searches. *IEEE Internet Computing , 4.1*, 34-43.

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., et al. (1999). LASSO: A Tool for Surfing the Answer Net. *Text Retrieval Conference (TREC-8).*

Möller, K., Ambrus, O., Dragan, L., & Handschuh, S. (2008). A Visual Interface for Building SPARQL Queries in Konduit. *7th International Semantic Web Conference (ISWC 2008).* Karlsruhe, Germany.

Montague, M., & Aslam, J. A. (2001). Metasearch consistency. *24th Annual International ACM SIGIR Conference on Research and De-velopment in Information Retrieval (SIGIR 2001)*, (pp. 386-387). New Orleans, Louisiana, USA.

Montague, M., & Aslam, J. A. (2001). Relevance score normalization for metasearch. *10th International Conference on Information and Knowledge Management (CIKM 2001)*, (pp. 427-433). Atlanta, Georgia, USA.

Montaner, M., Lopez, B., & De la Rosa, J. L. (2004). A Taxonomy of Recomended Agents on the Internet. *Artificial Intelligence Review* , 285-330.

Motta, E., & Sabou, M. (2006). Next Generation Semantic Web Applications. *1st Asian Semantic Web Conference.* Beijing, China.

Motta, E., Margas-Vera, M., Domingue, J., Lanzoni, M., Stutt, A., & Ciravegna, F. (2002). MnM: Ontology-driven semi-automatic and automatic support for semantic markup. *13th International Confererence on Knowledge Engineering and Knowledge Managment (EKAW02)*, (pp. 379-391). Siguenza, Spain.

Ng, K. B., & Kantor, P. B. (2000). Predicting the effectiveness of naive data fussion on the basis of system characteristics. *Journal of the American Society for Information Science* , 1177-1189.

Paice, C. D. (1991). A thesaural model of information retrieval. *Information Processing and Management 27* , 433-447.

Paslaru, E. (2005). Using Context Information to Improve Ontology Reuse. *Doctoral Workshop at the 17th Conference on Advanced Information Systems Engineering CAiSE'05.* Porto, Portugal.

Passin, T. B. (2004). *Explorer's Guide to the Semantic Web.* New York, NY, USA.: Manning Publications.

Pennock, D., Horvitz, E., & Giles, C. L. (2000). Social choice theory and recommender Systems: Analysis of the axiomatic foundations of collaborative filtering. *17th National Conference on Artificial Intelligence (AAAI 2000)*, (pp. 729-734). Austin, Texas, USA.

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. *21st annual international ACM SGIR conference on Research and development in information retrieval*, (págs. 275-281). Melbourne, Australia .

Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., & Kirilov, A. (2004). KIM – A Semantic Platform for Information Extraction and Retrieval. *Journal of Natural Language Engineering 10, Issue 3-4, Cambridge University Press* , 375-392.

Porzel, R., & Malaka, R. A. (2004). A task-based approach for ontology evaluation. *Workshop on Ontology Learning and Population. 16th European Conference on Artificial Intelligence. (ECAI 2004).* Valencia, Spain.

Prud'hommeaux, E., & Seaborne, A. (2006). *SPARQL Query Language for RDF.* W3C Working Draft.

Rau, L. (1987). Knowledge organization and access in a conceptual information system. *Information Processing and Management 23, Issue 4* , 269-283.

Renda, M. E., & Straccia, U. (2003). Web metasearch: rank vs. score based rank aggregation methods. *ACM symposium on Applied Computing*, (pp. 841-846). Melbourne, Florida, USA.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186). North Carolina, United States: ACM New York, USA.

Richardson, R., & Smeaton, A. (1995). sing WordNet in a knowledge-base approach to Information Retrieval. *BCS-IRSG Colloquium on Information Retrieval.*

Rijsbergen, C. J. (1979). *Information Retrieval.* London: Butterworths.

Robertson, S. E., & Sparck Jones, K. (1976). Simple, Proven Approaches to Text Retrieval. *Journal of the American Society for Information Science* , 129-146.

Rocha, C., Schwabe, D., & Aragão, M. P. (2004). A Hybrid Approach for Searching in the Semantic Web. *13th International World Wide Web Conference (WWW 2004)*, (pp. 374-383). NY.

Ruiz, M., Alfonseca, E., & Castells, P. (2007). Automatising the Learning of Lexical Patterns: an Application to the Enrichment of WordNet by Extracting Semantic Relationships from Wikipedia. *Data and Knowledge Engineering* , 484-499.

Sabou, M., Gracia, J., Angeletou, S., D'Anquin, M., & Motta, E. (2007). Evaluating the Semantic Web: A Task-based Approach. *6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference.* Busan, Korea.

Salton, G. (1986). *Introduction to Modern Information Retrieval.* New York, NY, USA: McGraw-Hill.

Salton, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing.* Upper Saddle River, NJ, USA: Prentice-Hall.

Sanderson, M. (1994). Word Sense Disambiguation and Information Retrieval. *17th annual international ACM SIGIR conference on Research and development in information retrieval.*

Saracevic, T., & Kantor, P. (1998). A study of information seeking and retrieving – III. Searchers, searches, overlap. *Journal of the American Society for Information Science* , 197-216.

Savoy, J., Le Calvé, A., & Vrajitoru, D. (1996). Report on the TREC-5 Experiment: Data Fusion and Collection Fusion. *5th Text REtrieval Conference (TREC 5)*, (pp. 489-502). Gaithersburg, Maryland, USA.

Seaborne, A. (2004). *RDQL – A Query Language for RDF.* W3C Member Submission.

Shah, U., Finin, T., Joshi, A., Cost, R., & Mayfield, J. (2003). Information Retrieval on the Semantic Web. *10th International Conference on Information and Knowledge Management.* ACM Press.

Shaw, J. A., & Fox, E. A. (1994). Combination of multiple searches. *3rd Text REtrieval Conference (TREC 3)*, (pp. 105-108). Gaithersburg, Maryland,USA.

Shaw, W. M., Burgin, J. R., & Howell, P. (1997). Performance Standards and Evaluation in IR Test Collections: Vector-Space and Other Retrieval Models. *Information Processing & Management* , v33 n1 p15-36.

Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., & Warke, Y. (2002). Managing Semantic Content for the Web. *IEEE Internet Computing 6, Issue 4* , 80-87.

Shoval, P. (1981). Expert/consultation system for a retrieval data-base with semantic network of concepts. *4th Annual International ACM SIGIR conference on Information storage and retrieval: theoretical issues in information retrieval*, (pp. 145-149). Oakland, CA.

Shuang, L., Fang, L., Clement, Y., & Weiyi, M. (2004). An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. *27th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 266-272). Sheffield, United Kingdom: ACM Press.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted Document Length Normalization. *19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 96)*, (pp. 21-29). Zurich, Switzerland.

Snášel, V., Moravec, P., & Pokorný, J. (2005). Using BFA with WorldNet ontology based model for Web retrieval. *Proceedings of the First IEEE International Conference on Signal-Image Technology & Internet-Based Systems (SITIS'05)*, (pp. 254-259). Yaoundé, Cameroon.

Snásel, V., Moravec, P., & Pokorný, J. (2005). WordNet Ontology Based Model for Web Retrieval. *International Workshop on Challenges in. WIRI 2005*, (pp. 220-225). Japan.

Spärck Jones, K. (2003). Document Retrieval: Shallow Data, Deep Theories, Historical Reflections, Potential Directions. *25th European Conference on Information Retrieval Research (ECIR 2003). 2633.* Pisa, Italy: Springer Verlag.

Spärck Jones, K. (1964). *Synonymy and Semantic Classification. Ph.D. thesis.* University of Cambridge, UK.

Sparck Jones, K., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management* , 779 - 808.

Srihari, K., Li, W., & Li, X. (2004). Information Extraction Supported Question- Answering. *In Advances in Open- Domain Question Answering* .

Staab, S., & Studer, R. (2004). Handbook on Ontologies. Berlin Heidelberg New York: Springer Verlag.

Stojanovic, N. (2003). On Analysing Query Ambiguity for Query Refinement: The Librarian Agent Approach. *22nd International Conference on Conceptual Modeling. 2813*, pp. 490-505. Berlin Heidelberg: Springer Verlag.

Stojanovic, N., Studer, R., & Stojanovic, L. (2003). An Approach for the Ranking of Query Results in the Semantic Web. *2nd International Semantic Web Conference (ISWC2003). 2870*, pp. 500-516. Berlin Heidelberg: Springer Verlag.

Sure, Y., & Iosif, V. (2002). First Results of a Semantic Web Technologies Evaluation. *Common Industry Program at the federated event: ODBASE'02 Ontologies, Databases and Applied Semantics.* California, Irvine.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., & Wenke, D. (2002). OntoEdit: Collaborative Ontology Development for the Semantic Web. *first International Semantic Web Conference 2002 (ISWC 2002).* Sardinia, Italy.

Taylor, P. (2007). New tools to vie with Google. *Financial Times* .

Tejedor, J., García, R., Fernández, M., López, F. J., Perdrix, F., Macías, J. A., et al. (2007). ntology-Based Retrieval of Human Speech. *6th International Workshop on Web Semantics (WebS 2007) at the 18th International Conference on Database and Expert Systems Applications (DEXA 2007).* Regensburg, Germany.

Todorov, D., & Schandl, B. (2008). *Small-Scale Evaluation of Semantic Web-based Applications.* Vienna, Austria: Department of Distributed and Multimedia Systems. University of Vienna.

Tsinaraki, C., Polydoros, P., Kazasis, F., & Christodoulakis, S. (2005). Ontology-Based Semantic Indexing for MPEG-7 and TV-Anytime Audiovisual Content. *Multimedia Tools and Applications , 26*, 299-325.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., et al. (2006). Semantic annotation for knowledge managment: Requirements and survey of the state of the art. *Journal of Web Semantics* , 14-28.

Van Rijsbergen, C. J. (1979). *Information Retrieval.* Butkrworthe, London.

Velardi, P., Navigli, R., Cuchiarelli, A., & Neri, F. (2005). Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. In P. Buitelaar, P. Cimiano, & B. Magnini, *Ontology Learning from Text: Methods, Evaluation and Applications.* Amsterdam, The Netherlands: IOS Press.

Vogt, C. C., & Cottrell, G. (1999). Fusion via a linear combination of scores. *Information Retrieval* , 151-173.

Vogt, C. C., & Cottrell, G. (1998). Predicting the Performance of Linearly Combined IR Systems. *21st ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 98)*, (pp. 190-196). Melbourne, Australia.

Vogt, C. C., Cottrell, G., Belew, R. K., & Bartell, B. T. (1996). Using relevance to train a linear mixture of experts. *5th Text REtrieval Conference (TREC 5)*, (pp. 503-516). Gaithersburg, Maryland, USA.

Voorhees, E. (2001). The Philosophy of Information Retrieval Evaluation. *Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems* (pp. 355-370). London, UK: Springer Verlag.

Voorhees, E., & Harman, D. K. (2000). Overview of the 9th TREC conference. *9th Text REtrieval Conference (TREC 2000)*, (pp. 1-14). Gaithersburg, Maryland, USA.

Vorhees, E. (1994). Query expansion using lexical semantic relations. *17th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 61-67). Dublin, Ireland: Springer-Verlag.

Vorhees, E. (2001). The TREC question answering track. *Natural Language Engineering 7(4)* , 361-378.

Vorhees, E. (1993). Using WordNet to Disambiguate Word Sense for Text Retrieval. *16th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 171-180). Pittsburgh, Pennsylvania, United States: ACM Press, New York, NY.

Wilks, Y. A., & Tait, J. I. (2005). A Retrospective View of Synonymy and Semantic Classification. In *Charting a New Course: Natural Language Processing and Information Retrieval.: Essays in Honour of Karen Spärck Jones* (pp. 1-11). Springer Netherlands.

Yang, Y., & Chute, C. G. (1993). Words or concepts: The features of indexing units and their optimal use in information retrieval. *17th Annual Symposium on Computer Applications in Medical Care. Washington*, (pp. 685-689). D. C., NY.

Zhang, L., Yu, Y., Zhou, J., Lin, C., & Yang, Y. (2005). An enhanced model for searching in semantic portals. *Proceedings of the 14th International World Wide Web Conference.* Chiba, Japan.