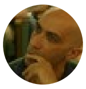
Knowledge extraction based on discourse representation theory and linguistic frames

CONFERENCE PAPER · OCTOBER 2012

DOI: 10.1007/978-3-642-33876-2_12

Valentina Presutti
Italian National Research Council
59 PUBLICATIONS **343** CITATIONS



Aldo Gangemi
Université Paris 13 Nord / ISTC-CNR
157 PUBLICATIONS **3,983** CITATIONS

Knowledge Extraction based on Discourse Representation Theory and Linguistic Frames

Valentina Presutti, Francesco Draicchio, and Aldo Gangemi

STLab, ISTC - CNR, Roma, Italy
{valentina.presutti,aldo.gangemi}@cnr.it,
draicchi@cs.unibo.it
http://stlab.istc.cnr.it

Abstract. We have implemented a novel approach for robust ontology design from natural language texts by combining Discourse Representation Theory (DRT), linguistic frame semantics, and ontology design patterns. We show that DRT-based frame detection is feasible by conducting a comparative evaluation of our approach and existing tools. Furthermore, we define a mapping between DRT and RDF/OWL for the production of quality linked data and ontologies, and present FRED, an online tool for converting text into internally well-connected and linked-data-ready ontologies in web-service-acceptable time.

Keywords: frame detection, discourse representation theory, robust ontology design, knowledge extraction

1 Introduction

The problem of knowledge extraction from text is still only partly solved, in particular if we consider it as a means for populating the Semantic Web. Being able to automatically and fastly produce quality linked data and ontologies from an accurate and reasonably complete analysis of natural language text would be a breakthrough: it would enable the development of applications that automatically produce machine-readable information from Web content as soon as it is edited and published by generic Web users, e.g. through a content management system plugin for automatically annotating HTML pages with RDFa on-the-go.

This simple vision is still far from being reached with state of the art tools. Approaches that tackle this task are referred to in the literature as *ontology learning and population* (OL&P) [8]. Ontology learning can be described as “the acquisition of a domain model from data”, hence it targets TBox production. Typical ontology learning tasks include concept extraction, relation extraction, and taxonomy induction. Learning the extension of concepts and relations (individuals and facts) is the typical task of ontology population, hence targeting ABox production. OL&P is meant to support ontology engineers in defining the appropriate ontology and filling its knowledge base according to the context given by a corpus of documents covering a certain knowledge domain.

In the context of knowledge management for large organizations, existing OL&P approaches can be used as drafting ontology engineering tools, but they show some limitations when used to produce linked data on the Web: they usually need a training phase, which can take a long time; their output form needs further elaboration to be put in a logical form; they do not exploit full OWL expressivity, since they typically focus on specific aspects e.g. taxonomy creation, disjointness axioms, etc.; in most cases they lack implementation of ontology design good practices; linking to existing linked data vocabularies and datasets is usually left as a separate task or not considered at all. In other words, existing tools focus mainly on the needs of (possibly large) organizations, where users would substantially refine the resulting ontology, as opposed to focusing on producing ontologies and linked data for the Web. The current trend to abridge organizational knowledge (specially from public administrations) with public semantic datasets is an additional motivation for handling OL&P in a way that works well with Semantic Web.

An important aspect of ontology learning is the design quality of the resulting ontology: this is related to representing the results in a logical form such as OWL by ensuring that modeling good practices are followed. Approaches such as [3] show that augmenting the learning cycle with selection and reuse of ontology design patterns, or with the detection of linguistic frames, improves results of existing OL methods.

Based on the above considerations, we summarize a set of requirements for a method that enables robust OL&P, the output of which can be readily published on the Web:

- ability to capture accurate semantic structures, and to produce good quality schemas e.g. representing complex relations;
- exploitation of general purpose resources i.e. no need of large-size domain-specific text corpora and training sessions;
- minimal time of computation;
- ability to map natural language to RDF/OWL representations;
- easiness of adaption to the principles of linked data publishing [15].

We present a novel approach and an online tool, FRED,[1], which performs robust OL&P according to those requirements: FRED performs deep parsing of natural language and extracts complex relations based on Discourse Representation Theory (DRT) [16]. We use Boxer [5] that implements a DRT-compliant deep parser, which saves FRED from the typical training phase of machine-learning-based information extraction tools. We extend Boxer in order to detect the most appropriate linguistic frames [21] capturing complex relations expressed in the input text. This ensures good quality of design of the resulting ontology (cf. [10]). The logical output of Boxer with frames is transformed into RDF/OWL by means of a mapping model and a set of heuristics that follow good practices of OWL ontologies and RDF data design, e.g. we avoid blank nodes and create domain-oriented relation names. The produced RDF complies with linked data principles as we reuse existing vocabularies when possible, resolve named entities over resources existing in RDF datasets of the linked data cloud (LOD)[2], and

[1] FRED demonstrator is available at `http://wit.istc.cnr.it/fred`

[2] Named entity resolution relies on an external system, as described in Section 5

disambiguate domain terminology against WordNet and foundational ontologies. We are able to prove that our method has high potential for being the enabler of robust knowledge extraction, since it guarantees good quality of design, and fast computational performance.

The paper is structured as follows: Section 2 discusses related work. Section 3 shows a new approach to frame detection and its evaluation, Section 4 discusses a set of heuristics that we have defined for transforming a DRT-based logical form into a well designed RDF/OWL ontology, and Section 5 describes FRED, the system that implements the overall method. In Section 6 we briefly conclude and discuss future work.

2 Related Work

OL&P is concerned with the (semi-)automatic generation of ontologies from textual data (cf. [8]). Typical approaches to OL&P are implemented on top of Natural Language Processing (NLP) techniques, mostly machine learning methods, hence they require large corpora, sometimes manually annotated, in order to induce a set of probabilistic rules. Such rules are defined through a training phase that can take long time. OL&P systems are usually focused on either ontology learning (OL) for TBox production, or ontology population (OP) for ABox production.

Examples of OL systems include [17], which describes an ontology-learning framework that extends typical ontology engineering environments by using semiautomatic ontology-construction tools, and Text2Onto [9], which generates a class taxonomy and additional axioms from textual documents.

Examples of OP systems include: [23], which presents a GATE plugin that supports the development of OP systems and allows to populate ontologies with domain knowledge as well as NLP-based knowledge; [22] describes a weakly supervised approach that populate an ontology of locations and persons with named entities; [18] introduces a sub-task of OP restricted to textual mentions, and describes challenging aspects related to named entities. More complete reviews of state-of-art research methods and tools for OL&P are given in [8, 24, 13].

The method and tool (FRED) that we present in this paper differs from most existing approaches, because it does not rely on machine learning methods. Instead, it is based on a logical interpretation of natural language given by Discourse Representation Theory, a formal theory of linguistic semantics originally designed by Hans Kamp to cope both with linguistic phenomena – such as donkey sentences, anaphoric resolution, ellipsis and presupposition – and temporal relations [16]. Another distinguishing component of our approach is frame-based ontology design combined with a set of mapping and heuristic rules. We argue that automatic frame-based ontology design simulates the typical approach of ontology engineers when using textual documents as requirements for the ontology to be designed. Our method addresses the overall conceptualization expressed by a document, instead of only focusing on specific tasks such as named entity recognition or taxonomy induction.

We use Boxer [5] as an implementation of DRT, and we prove, by comparing it to Semafor [7], that it can be used for detecting frames with good performance, especially in terms of computational time. The choice of performing frame-based ontology design is based on the evidence given in [3] that OL methods performances improve if the learning cycle is augmented with ontology design patterns, and on the work by [10, 19] that ontology design patterns can be easily derived from frames. Additional related work on using frame semantics when integrating structured knowledge in NLP can be found in [20]. A notable work, which is also an inspiration for FRED, is AURA [6], which uses a library of frame-like knowledge engineering components (CLib) for automatic formalization of specialized texts into the KM language.

The core of the work on FRED presented in this paper was originally developed as a master's thesis [12]. Approximately in the same period, an approach similar to ours, named LODifier [1], has been developed. Unfortunately, we could not compare the performances of our system to LODifier's because the latter is not available at this time. Nevertheless, from the paper we could understand the main differences. LODifier reuses Boxer, however it does not exploit it as a frame-detector and does not follow a frame-based approach to ontology design. Their basic conversion table from DRT to RDF is similar to FRED's, but the result of Boxer is serialized to RDF without applying specific rules to maximizing design choices according to Semantic Web and OWL ontology design principles (e.g. it produces blank nodes for all variables and DRSs, does not consider terminological issues, does not try to reduce redundant structures, etc.). In our opinion, although it is in general recommended to publish linguistic linked data without changing the original data structure significantly, in the case of Boxer, we are not publishing established lexical data, but extracted knowledge in logical form that is not targeted at linguists, therefore there is no advantage in preserving the original data structure. LODifier makes a smart usage of NER and WSD (Word Sense Disambiguation) to WordNet to provide better linked data. For NER, we use Stanbol enhancers[3], and for WSD we have reused UKB[4] in a similar way as LODifier does, but also adding top-level mappings to DOLCE+DnS foundational ontology and to WordNet "super senses" (see Section 5). Our frame-based tools can be tested online.[5].

3 DRT-based frame detection

Robust OL&P requires special attention to design quality. [3] proves that augmenting the learning cycle with ontology design patterns improves existing OL&P tool performances in terms of ontology enrichment. [10] shows that detecting the most appropriate frames from the input text leads to improving the design quality of the resulting ontology because frames can be directly mapped to an

[3] Cf. `http://incubator.apache.org/stanbol/docs/trunk/enhancer/`

[4] `http://ixa2.si.ehu.es/ukb/`

[5] FRED is cited; for NER and WSD over FRED see our Wikipedia typer *Tìpalo* at `http://wit.istc.cnr.it/tipalo`.

important variety of ontology design patterns (cf. [19]) based on *n-ary relations*, the most critical logical form used in domain ontologies. We take these results as assumptions in our work, and design our workflow so that it includes a frame detection step that we use as a means for selecting ontology modeling choices.

Frames. Frame Semantics [14] is a formal theory of meaning: its basic idea is that humans can better understand the meaning of a single word by knowing the contextual knowledge related to that word. For instance, the sense of the word *buy* is clarified by knowing about the context of a commercial transfer that involves certain individuals, e.g. a seller, a buyer, goods, money, etc. Linguistic frames are referred to by sentences in text that describe different situations of the same type i.e. frame occurrences. The words in a sentence "evoke" concepts as well as the perspective from which the situation expressed is viewed.

In the previous example, the word *sell* evokes a situation from the perspective of the seller, and the word *buy* evokes it from the perspective of the buyer. This fact explains the observed asymmetries in many lexical relations that need a particular design involving roles in order to be represented. Frame semantics allows real-world knowledge to be captured by semantic frames, which describe particular types of situations, objects or events, and their participants characterized by specific semantic roles. FrameNet[21] is a lexical resource that collects linguistic frames, each described with its semantic roles, called frame elements, and lexical units (the words evoking a frame).

Frame detection. The frame detection (or frame recognition) task [10] has the goal of recognizing complex relations in natural language text. There are a number of systems that perform frame detection with reasonable performances, however they all require a training phase and the availability of a large annotated corpus. One of our goals is to realize a system that can be used also in interactive applications, in other words it has to be as fast and simple as possible.

A problem of machine learning-based systems is that their output needs significant intervention in order to be transformed into a logical form. This is an issue in our case, as we want to reuse the frame structure (e.g. frame roles) in order to reflect it in the ontology we produce. That is why the detection task per-se addresses only partially our requirements.

In summary, we need to answer the following questions:

1. How can we map natural language to a logical form?
2. How can we perform frame detection without ad-hoc training?

3.1 Discourse Representation Theory and Boxer

The answer to the first question is "Discourse Representation Theory" (DRT). DRT is a formal theory of meaning originally described in [16], and is equivalent to first-order logic (FOL). DRT uses an explicit semantic structured language called Discourse Representation Structure (DRS): standard representations corresponding to natural language sentences, which constitute the core of DRT languages. The flavour of DRT we are interested in provides an event-based, Neo-Davidsonian (based on reified n-ary relations just as frames are) model to represent natural language.

Boxer [5], an implementation of compositional semantics of language that produces a DRS output, is especially suited to our task. It is open-source software that performs deep parsing of natural language: it uses Combinatory Categorial Grammar (CCG) and produces event-based, verb-centric, semantic representations of natural language complying with DRT semantics. These representations are expressed in the form of DRS using the VerbNet[6] inventory of thematic roles. Boxer provides us with an important component of our workflow: it produces, without any additional training, a logical representation of natural language text. The fact that Boxer exploits VerbNet is helps answering our second question: since VerbNet is linked to FrameNet [21], we exploit such mappings in order to perform frame-detection without any training.

3.2 Using Boxer as a Frame Detection System

All frame detection systems address their task with a probabilistic approach. The formal structure of a FrameNet frame (defined at frame-creation time) can be defined as its proper semantic footprint [21]. Based on this observation, we had the intuition that a frame can be detected with a different approach from the probabilistic one: a rule-based approach whose output can be compared with the syntactic and semantic structure of frames, i.e. their typical syntactic manifestation in language, and with the involved roles that characterize them, in order to identify the best frame candidates.

This intuition can be experimented by using Boxer with a different purpose than its usual one, and the potential of the approach can be evaluated by comparing its performances against Semafor [11], to our knowledge the best performing tool for frame detection so far.

Boxer exploits VerbNet in order to identify the roles involved in a sentence, and roles in turn are used to detect a corresponding frame. However, as it is not developed with this task in mind, its coverage with respect to FrameNet frames is limited. Hence, we have adapted Boxer for tackling the frame detection task by integrating it with a resource that provides the most complete mapping between VerbNet and FrameNet.[7].

Evaluation. In order to evaluate Boxer as a frame detection tool, we compare its performances against Semafor's for *Task 19* of *Semeval'07* [2], defined as a *frame recognition task*: given a textual sentence, the system has to automatically extract facts from it, and predict FrameNet frame structures that best fit those facts. Although Semeval provides a set of benchmarks for evaluating the results of the tests, we could not use them because Semafor has been trained on their annotated corpus after the challenge.

[6] VerbNet, `http://verbs.colorado.edu/~mpalmer/projects/verbnet.html`

[7] `http://verbs.colorado.edu/~mpalmer/projects/verbnet/downloads.html`

Tool	Precision	Recall	F-Score	Coverage	Elapsed Time
Boxer	69.693	53.223	60.354	76.367	2m 45.21s
SEMAFOR	72.370	71.875	72.122	99.316	20m 14.27s

Table 1: Performances for frame recognition task

To address this problem, we have built a new benchmark[8] that is based on the FrameNet-annotated corpus of sample frame sentences. The dataset consists of 1214 sentences, each fully annotated with at least one frame. The benchmark is automatically generated by randomly selecting dataset sentences among the whole set of annotated samples so as to keep the same data distribution. The evaluation is performed in terms of precision, recall, coverage, and time-efficiency. We have built an evaluation component by implementing the same criteria used in the Semeval task evaluation. Such component is used for comparing the results of the two systems against the gold-standard file provided by the benchmark. Given a sentence, when a predicted frame matches exactly the one indicated by the gold-standard, we assign a full score (1). Additionally, by following criteria defined in [3], when a mismatch occurs, we assign a partial score (0.8) if the predicted frame is conceptually close, or semantically related to the one indicated by the gold-standard. For the sake of this evaluation we have used Semafor for frame detection only from verbs, because Boxer currently identifies frames expressed by verbs. While this is surely a gap in frame detection, consider that the explicit roles for non-verbal frames are usually very limited.

Table 1 shows the results for the frame recognition task performed by Boxer and Semafor by considering only exact match (full scores). Precision, recall, f-score and coverage values are expressed in percentage, while the computation time is expressed in minutes and seconds. From the results, we notice that Boxer is much faster than Semafor, however the latter still performs better than Boxer in terms of accuracy, although precision values are comparable. We claim that this is a good result as the goal of our evaluation is to understand the feasibility of the rule-based approach for frame detection. Our focus has been so far the adaptation of Boxer to the frame detection task (by enriching its reference knowledge base through the integration of Semlink mappings), hence we are aware that there is room for improving the detection algorithm. We leave this issue to future work that includes the implementation of a frame disambiguation method; at the moment Boxer selects the first candidate among a list of frames that it identifies as possible targets without performing any ranking.

While precision values are comparable, we observe a significant gap in recall, which is mainly due to the difference in coverage of the two considered systems. While Boxer covers around 76% of answers, Semafor scores 99%. This difference can be further analyzed by considering performance with partially correct answers (partial scores), reported in Table 2.

In this case, we notice a substantial increase of Boxer performances over all the evaluation parameters, while Semafor shows a less significant improvement. An important result of this evaluation is that a rule-based approach can easily

[8] The benchmark is available online at `http://tinyurl.com/fd-benchmark`

Tool	Precision	Recall	F-Score
Boxer	75.320	57.519	65.227
Semafor	75.325	74.797	75.060

Table 2: Performances for frame recognition task taking into account partially correct results

reach performances that are comparable to those of the best frame detection tool currently available based on a probabilistic approach. We claim that these results are promising, they demonstrate the high potential of our method, especially if we consider that the detection algorithm can be further improved by enhancing it for frame disambiguation, and by further extending the set of Semlink mappings between VerbNet and FrameNet.

A number of additional consideration can be made for supporting our claim: Boxer frame detection system is indirectly related to FrameNet, hence it currently exploits FrameNet knowledge only partially. As future work, we aim at improving this aspect by exploiting the FrameNet-LOD datasets [19], which encodes the full knowledge of FrameNet including semantic relations between frames.

Probabilistic models such as those used by Semafor encode most of the knowledge expressed by FrameNet as a consequence of being trained over a large portion of FrameNet data. In other words, the scope of information used for building the inductive models in Semafor is almost complete with respect to FrameNet knowledge. This is the main motivation for the reduced coverage of Boxer, and therefore also of the high amount of partially correct answers. We expect that by increasing Boxer's coverage, its accuracy will increase as well. Two further important aspects are: time taken to compute predictions and output form. Figure 1 shows that Boxer is much faster than Semafor, as also reported in Table 1. This is probably due to the fact that Semafor algorithm has a very expensive complexity: it also requires significant resources (8 GB RAM and CPU cycles). As far as the output form is concerned, Semafor does not provide a logical representation, which is one of our core requirements.

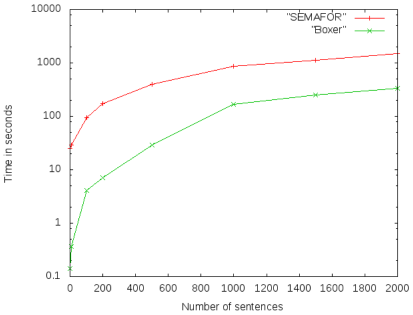


Fig. 1: Time taken to provide answers in function of the number of sentences per document.

Finally, we remark that the benchmark used here put Semafor in a condition of advantage with respect to Boxer. The benchmark is built on FrameNet sample sentences, and Semafor is trained over FrameNet full text annotations. If we would use Semafor on a corpus independent on FrameNet, it would need a proper training, while Boxer could be directly executed without any preparing activity. These characteristics, and the good performances of Boxer support our claim of its suitability for performing frame detection on any Web content item, and for being employed in the context of interactive Semantic Web applications.

4 Transforming DRT forms to OWL/RDF ontologies

A key step in our OL&P approach is performing good quality ontology design. We assume, supported by [10, 3], that a frame-based approach to design helps quality assurance. Intuitively, a frame provides a means for representing knowledge boundaries, which is a desideratum, since it allows to associate a context to data in a knowledge base. In this work, we consider frames identified by verbs, which is also the case in most modeling situations. For example, if an ontology engineer needs to store in a knowledge base the knowledge expressed by the following sentence:

> The statement by China Foreign Ministry on Friday signaled a possible breakthrough in a diplomatic crisis that has threatened American relations with Beijing.[9]

she would probably model at least two situations (or events): one expressed by the verb *signal*, the other expressed by the verb *threaten.* The two situations create boundaries for the *statement*, its content, and the *Ministry*, and link them to *diplomatic crisis*, *America*, and *Beijing*, which in turn are kept together by the *threaten* situation.

Thanks to the frame-based approach integrated in our method, as described in Section 3, we can automatically design an ontology by following this good modeling practice. However, although Boxer gives us a logical form, its constructs – syntactically, lexically, and semantically – differ from RDF or OWL ones, and the heuristics that it implements for interpreting a natural language and transforming it to a DRT-based structure can be sometimes awkward when directly translated to ontologies for the Semantic Web.

In this section we show, through a set of examples, the rules that we have defined in order to transform Boxer output to an OWL/RDF ontology. We can distinguish two types of rules: (i) translation rules, which define global transformations from DRS constructs to OWL constructs, and (ii) heuristic rules, which define local transformations that deal with adapting the results of Boxer heuristics to the needs of a Semantic Web ontology.

Translation rules. DRT is basically FOL (although Boxer uses a subset of it), hence the first step is to define a set of global translation rules that allow us to

[9] Taken from the New York Times, May 4th 2012, `http://www.blogrunner.com/snapshot/D/7/2/chen_guangcheng_can_apply_to_study_abroad_china_says/`.

transform DRS constructs into OWL/RDF constructs (except when overruled by local rules, see below). Table 3[10] indicate DRT constructs, their syntax in Boxer, their corresponding FOL construct, and their corresponding OWL/RDF construct. Additionally, Boxer has a set of built-in predicates. Those most frequently used are listed in Table 4,[11] each associated with a Semantic Web entity, to which we align it by default. We have represented all Boxer built-in types and relations in a publicly available ontology[12]. Finally, the semantic roles that are applied to the frames detected in sentences can belong to three different vocabularies: the default one is VerbNet, the second is FrameNet, the third is a domain relation that cannot be resolved to any of VerbNet or FrameNet roles.[13]

DRT construct	**Boxer syntax**	**FOL construct**	**OWL construct**
Predicate	pred(x)	Unary predicate ϕ	`rdf:type`
Relation	rel-name(x,y)	Binary relation	`owl:ObjectProperty`
Eq Rel	eq(x,y)	Identity	`owl:sameAs`
Named Entity	named(<var>, <name>, <type>)	Unary predicate ϕ	`owl:NamedIndividual`
Discourse Referent	(<var>)	Quantified Variable	(generated) `owl:NamedIndividual`
DRS	<drs> with event E	Proposition P with predicate ϕ_E	RDF graph G_P with class `E`
Negated DRS	not(<drs>)	Negated Proposition $\neg P$	G_P with `NotE` `owl:disjointWith` `E`

Table 3: The main translation rules from DRS to OWL.

Boxer built-in type	**Label**	**Semantic Web entity**
Per	Person	foaf:Person
Org	Organisation	foaf:Organisation
Loc	Location	dbpedia:Place
Tim	Time	to:Interval
Ttl	Title	dul:Role
Event	Event	dul:Event
Eq	Equal to	owl:sameAs

Table 4: Boxer built-in types and relations.

For example, the sentence "Paul Newman hit the window with an open hand" is transformed by Boxer in the "boxed" form shown in Example 1. Each box represents a DRS: at the top it shows its variables, at the bottom its formal sentences. Boxer recognizes the roles `agent`, `patient`, and `instrument` and uses them to link entities to the event `x2` denoted by the verb *hit*, which expresses the situation type `hit` occurring in the sentence. It recognizes that the agent `x0` is a named entity `paul_newman` and that he is a person (`per`); that the patient `x1` is a `window` and the instrument `x3` is a `hand` that is also `open` (by co-reference of the

[10] NotE and E are generated classes for the event E contained in a (simple) DRS. The prefixes map to the following namespaces: rdf: `http://www.w3.org/1999/02/22-rdf-syntax-ns`, owl: `http://www.w3.org/2002/07/owl`

[11] With prefixes: foaf: `http://xmlns.com/foaf/0.1/`; dbpedia: `http://www.dbpedia.org/ontology/`, dul: `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#`, to: `http://www.w3.org/2006/time#`

[12] Boxer types and relation ontology, `http://www.ontologydesignpatterns.org/ont/boxer/boxer.owl`

[13] With prefixes: vn: `http://www.ontologydesignpatterns.org/ont/vn/abox/role/`; fn: `http://www.ontologydesignpatterns.org/ont/framenet/abox/fe/`; domain: <namespace chosen by the user>.

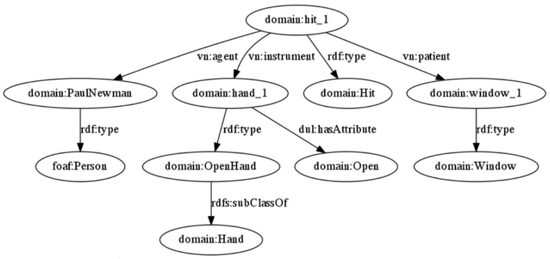


Fig. 2: FRED RDF graph for the sentence "Paul Newman hit the window with an open hand"

variable). All discourse referents (variables) here are existentially quantified: this interpretation always holds in DRT, except when a variable is in the antecedent DRS of an implication, so bearing a universal quantification.

```
Example 1
%%%   _________________________   _________________
%%%  |x0 x1                    | |x2 x3            |
%%%  |.........................| |.................|
%%% (|named(x0,paul_newman,per)|A|hit(x2)          |)
%%%  |window(x1)               | |Agent(x2,x0)     |
%%%  |_________________________| |Patient(x2,x1)   |
%%%                              |open(x3)         |
%%%                              |hand(x3)         |
%%%                              |Instrument(x2,x3)|
%%%                              |_________________|
```

Based on the mappings reported in Table 3 and Table 4, our tool FRED (that implements our method and is described in section 5) generates RDF code that produces e.g. the graph depicted in Figure 2. The resulting ontology defines the class `Hit`, which is a situation type i.e. a frame. Such situation type is instantiated by a named individual `hit_1`. The situation (frame occurrence) `hit_1` involves: (i) `PaulNewman`, an instance of `foaf:Person` having the role of `vn:agent` in the situation; (ii) an instance of `OpenHand` having the role of `vn:instrument` and an attribute `Open`; and (iii) an instance of `Window` having the role of `vn:patient`.

Most of the generated RDF works as from the translation rules shown in Table 3, but although this example is fairly simple, it makes it emerge more structure than those rules can generate. In fact, in order to design a proper OWL ontology, we need more design-oriented rules. The reason for additional rules lies in the way Boxer applies a "flat" FOL modeling style to natural language in order to produce a DRT-based logical form. FOL modeling style is not always compatible or appropriate to Semantic Web and Linked Data design. We cannot exemplify all additional heuristic rules implemented by FRED, however we provide some sample cases that demonstrate our approach.

Heuristic rules. In the simple example of Figure 2, we notice some non-trivial names and axioms, which cannot be derived from translation rules. For example, `open(x3)` and `hand(x3)` are only co-referential in Boxer, but do not form a unique term. A heuristical rule fires here based on the co-reference, and generates a new term for the class `OpenHand`. Moreover, an additional class `Hand` is created by FRED, and defined as super-class of `OpenHand`: this heuristic rule defines a default behavior when a branching structure is met in the text, in English this is a *left branching* (or "head-final") construction. Finally, the predicate `open` denoted by the adjective *open* is used by FRED as an attribute of `x3`.

Another heuristic rule has to do with naming. FRED uses the CamelCase convention as it is pretty popular on the Semantic Web. For example, properties start with lower case, while classes and individuals start with capital case.

An important issue is constituted by blank nodes, which are not desirable in linked data, but should be produced out of Boxer's variables. FRED implements a heuristics that creates individuals with a generated name to existentially quantified variables that are not resolved as named entities. In the sample graph from Figure 2, `hit_1` and `window_1` are such individuals.

Other design heuristics have to do with the generation of terminology associated with the definition of appropriate classes or properties. Let's consider the more complex sentence "At the meeting of European Union leaders, Germany leader Angela Merkel was facing Mario Monti, who forced the Iron Chancelor to blink.", in this case Boxer produces the result shown in Example 2.

```
Example 2
%%   -----------------------------   ----------------------------
%%  |x0 x1 x2 x3 x4               | |x5 x6 x7 x8                 |
%%  |.............................| |............................|
%% (|named(x0,angela_merkel,loc)  |A|force(x5)                   |)
%%  |named(x0,germany,loc)        | |face(x7)                    |
%%  |leader(x0)                   | |agent(x5,x1)                |
%%  |named(x1,mario_monti,per)    | |theme(x5,x6)                |
%%  |tenacious(x1)                | |     -----------------      |
%%  |opponent(x1)                 | |    |x9               |     |
%%  |named(x2,iron_chancellor,org)| | x6:|.................|     |
%%  |summit(x3)                   | |    |Body_movement(x9)|     |
%%  |meeting(x3)                  | |    |Agent(x9,x2)     |     |
%%  |named(x4,brussels,loc)       | |    |_________________|     |
%%  |_____________________________| |finally(x5)                 |
%%                                  |agent(x7,x0)                |
%%                                  |patient(x7,x1)              |
%%                                  |named(x8,european_union,org)|
%%                                  |leader(x8)                  |
%%                                  |of(x3,x8)                   |
%%                                  |in(x3,x4)                   |
%%                                  |at(x7,x3)                   |
%%                                  |____________________________|
```

On its turn, FRED produces the ontology depicted in Figure 3, in this case using FrameNet frames and roles. The power of the design heuristics is even more evident here. The preposition *of* is used for generating the property `domain:leaderOf` between instances of `Leader` (e.g. `AngelaMerkel`) and of `Organization` (e.g. `EuropeanUnion`). The situation `blink_1` with frame `fn:BodyMovement` is an argument to another situation `force_1` from frame `fn:ConfrontingProblem`. The definition of classes and properties guided by the occurrence of specific lexico-syntactic patterns is the subject of a number of FRED heuristics, all designed by respecting the requirement of preserving the frame-like structure of the designed ontology.

5 Prototype

Our method is implemented in a tool named FRED, which is accessible online.[14] Figure 4 shows FRED's architecture. FRED is designed in order to be deployed as a web service, hence one important goal it has to address is minimizing computing time. In addition it implements a modular, highly interoperable and customizable architecture in order to ensure reusability by other applications, and extensibility.

[14] `http://wit.istc.cnr.it/fred`

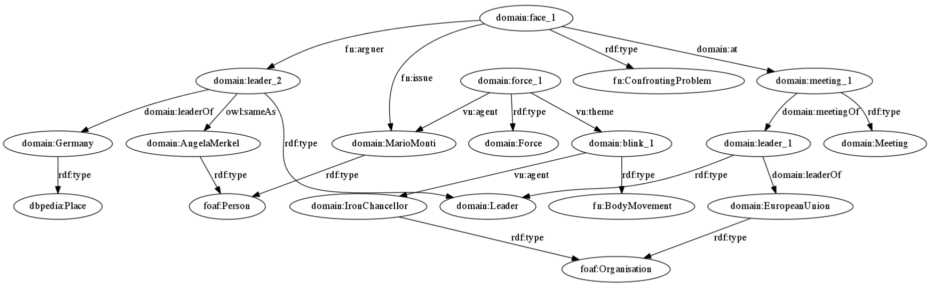


Fig. 3: FRED RDF graph for the sentence "At the meeting of European Union leaders, Germany leader Angela Merkel was facing Mario Monti, who forced the Iron Chancellor to blink."

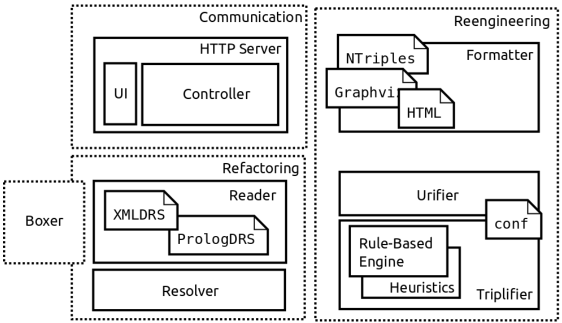


Fig. 4: FRED software architecture.

The framework is constituted by four main components: (i) *Boxer* (implemented in Prolog) performs deep parsing of natural language text including frame-detection, and provides an output a DRS output; (ii) the *communication* component realizes a lightweight HTTP server based on RESTful architecture. This component is in charge of publicly exposing APIs for querying the system. It takes a language text and some optional parameters as input, and returns an OWL/RDF ontology; (iii) The *refactoring* component transforms Boxer output in a form to be passed to the re-engineering component, which is responsible of implementing the semantic transformation from DRT to OWL; (iv) the *re-engineering* component implements all translation and heuristic rules described in Section 4. The last three components are implemented in Python.

In addition to such components, FRED exploits external services performing entity resolution on linked data, and word sense disambiguation. The NER component is named "Enhancer" and is part of the Apache incubating project "Stanbol"[15]. Stanbol allows to indicate any number of datasets to be used as sources for entity recognition and resolution, hence providing great flexibility and customizability with respect to the entities that are of interest for a FRED user. WSD is performed by means of the UKB tool[16], by aligning domain classes to WordNet synsets, and the synsets to DOLCE+DnS foundational ontology classes[17] and WordNet lexnames ("super-senses").[18]. Additional linking is pro-

[15] http://incubator.apache.org/stanbol/

[16] http://ixa2.si.ehu.es/ukb/

[17] http://www.ontologydesignpatterns.org/ont/dul/DUL.owl

[18] http://wordnet.princeton.edu/man/lexnames.5WN.html

vided for special purposes, for example the “Tìpalo” service[19] employs FRED and external services to automatically type the entities referred by Wikipedia pages.

6 Conclusion and future work

We have presented a novel approach and a tool, FRED, for ontology learning and population in the Semantic Web. It is based on DRT and frame-based ontology design. In order to demonstrate its potential, we have extended one of its core components, Boxer, to perform frame recognition, evaluating its performances against the state-of-art tool, Semafor. Results are promising: this new approach to frame detection shows better time computational performance than existing tools for frame detection, and the ontologies produced are internally well-connected and linked-data-ready. The real payoff of having an RDF and OWL representation of texts is in fact the ability to quickly and accurately process relevant texts for the population of the Semantic Web. FRED is able to semantically map DRT logical forms (DRS) to RDF and OWL by exploiting frame-based design, and by implementing a set of heuristics that address terminology and structure generation according to Semantic Web design practices.

Future work includes the improvement of the frame detection algorithm through e.g. extending Boxer coverage of FrameNet frames, and by addressing frame disambiguation. Additionally, we are performing an extensive evaluation of the overall workflow in terms of computing time and design quality of the produced ontologies. To this purpose, we have defined a user-based evaluation that involves expert ontology engineers in evaluating FRED on a set of preselected texts. The evaluation is conducted by adopting the quality measures and experimental settings that we have defined and executed in [4].

References

1. Isabelle Augenstein, Sebastian Padó, and Sebastian Rudolph., *LODifier: Generating Linked Data from Unstructured Text*, Extended Semantic Web Conference, Springer, 2012.
2. Collin F. Baker, Michael Ellsworth, and Katrin Erk, *Semeval'07 task 19: frame semantic structure extraction*, Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval '07, ACL, 2007, pp. 99–104.
3. Eva Blomqvist, *Ontocase-automatic ontology enrichment based on ontology design patterns*, International Semantic Web Conference, 2009, pp. 65–80.
4. Eva Blomqvist, Valentina Presutti, Enrico Daga, and Aldo Gangemi, *Experimenting with extreme design*, EKAW, 2010, pp. 120–134.
5. Johan Bos, *Wide-Coverage Semantic Analysis with Boxer*, Semantics in Text Processing (Johan Bos and Rodolfo Delmonte, eds.), College Publications, 2008, pp. 277–286.

[19] `http://wit.istc.cnr.it/tipalo`.

6. Vinay K. Chaudhri, Bonnie John, Sunil Mishra, John Pacheco, Bruce Porter, and Aaron Spaulding, *Enabling Experts to Build Knowledge Bases from Science Textbooks*, Proceedings of KCAP 2007, 2007.
7. Desai Chen, Nathan Schneider, Dipanjan Das, and Noah A. Smith, *Probabilist frame-semantic parsing*, Proceedings of NAACL-HLT, 2010.
8. Philipp Cimiano, *Ontology learning and population from text: Algorithms, evaluation and applications*, Springer, 2006.
9. Philipp Cimiano and Johanna Vlker, *Text2onto - a framework for ontology learning and data-driven change discovery*, 2005.
10. Bonaventura Coppola, Aldo Gangemi, Alfio Massimiliano Gliozzo, Davide Picca, and Valentina Presutti, *Frame detection over the semantic web*, ESWC (Lora Aroyo et al., ed.), LNCS, vol. 5554, Springer, 2009, pp. 126–142.
11. Dipanjan Das and Noah A. Smith, *Semi-supervised frame-semantic parsing for unknown predicates.*, ACL (Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, eds.), The Association for Computer Linguistics, 2011, pp. 1435–1444.
12. Francesco Draicchio, *Frame-driven Extraction of Linked Data and Ontologies from Text*, Master's Thesis, University of Bologna Electronic Press, February 2012, http://amslaurea.unibo.it/3165/.
13. Georgios Petasis et al., *Ontology Population and Enrichment: State of the Art*, Knowledge-Driven Multimedia Information Extraction and Ontology Evolution, LCNS, vol. 6050, Springer, 2011, pp. 134–166.
14. Charles J. Fillmore, *Frame semantics*, pp. 111–137, Hanshin Publishing Co., Seoul, South Korea, 1982.
15. Tom Heath and Christian Bizer, *Linked data: Evolving the web into a global data space (1st edition)*, Synthesis Lectures on the Semantic Web: Theory and Technology 1:1, Morgan & Claypool, 2011.
16. Hans Kamp, *A theory of truth and semantic representation*, Formal Methods in the Study of Language (Jeroen A. G. Groenendijk, Teo M. V. Janssen, and Martin B. J. Stokhof, eds.), vol. 1, Mathematisch Centrum, 1981, pp. 277–322.
17. Alexander Maedche and Steffen Staab, *Ontology learning for the semantic web*, IEEE Intelligent Systems **16** (March-April 2001), pp. 72–79.
18. Bernardo Magnini, Emanuele Pianta, Octavian Popescu, and Manuela Speranza, *Ontology Population from Textual Mentions: Task Definition and Benchmark*, Proc. of the 2nd Workshop on Ontology Learning and Population, ACL, 2006.
19. Andrea G. Nuzzolese, Aldo Gangemi, and Valentina Presutti, *Gathering Lexical Linked Data and Knowledge Patterns from FrameNet*, Proc. of the 6th International Conference on Knowledge Capture (K-CAP) (Banff, Alberta, Canada), 2011.
20. Ekaterina Ovchinnikova, *Integration of World Knowledge for Natural Language Understanding*, Atlantis Press, Springer, 2012.
21. Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Sceffczyk, *Framenet ii: Extended theory and practice*, 2010.
22. Hristo Tanev and Bernardo Magnini, *Weakly supervised approaches for ontology population*, Proceedings of the 2008 conference on Ontology Learning and Population, IOS Press, 2008, pp. 129–143.
23. René Witte, Ninus Khamis, and Juergen Rilling, *Flexible Ontology Population from Text: The OwlExporter*, LREC (Nicoletta Calzolari et al., ed.), European Language Resources Association, 2010.
24. Ziqi Zhang and Fabio Ciravegna, *Named Entity Recognition for Ontology Population using Background Knowledge from Wikipedia*, Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances (W. Wong, W. Liu, and M. Bennamoun, eds.), IGI Global, 2011.