

DART-2011

September 17, 2011 - Palermo (Italy)

From Logical Forms to SPARQL Query with GETARUN

Rocco Tripodi – Rodolfo Delmonte

Computational Linguistics Laboratory Ca' Foscari University

<http://project.cgm.unive.it/>



Outlines

Introduction

the Semantic Web and the automatic processing of information.

Referencing things in the real world.

The *common sense knowledge* and the *encyclopedia*.

SW and NLP

Problems

Technique

GETARUN

QA over Linked Data

Question Analysis

Logical Forms

SPARQL schemata

Examples

Results

The Semantic Web and the automatic processing of information

Nowdays, the need of the automatic processing of information on the web has become more and more relevant in order to develop applications able to cope with unstructured information.

We could consider the SW to have an Information Extraction task at its heart. The SW task consists in relating entities to specific categories (e.g. Person, Place, Event, etc.).

The referencing procedure developed over the years is based on metadata and ontology. The metadata provides a computer-readable concept specification and the ontology provides a conceptual knowledge structure, which organizes concepts.

The procedure of referencing things

The standard way to access a KB on the SW is SPARQL. SPARQL is used to express queries across data sources, whether the data is stored or viewed as RDF.

RDF facts have a simple syntax and could be easily used to compare facts in the text. Subject – Predicate – Object.

We consider the SW datasets as encyclopedia (understood within the meaning given by Umberto Eco, as a network of interconnected cultural units), where we could find information about entities, and the reference could be considered as an attribution of meaning.

We could use the SW data to:

- organize the information contained in the text.

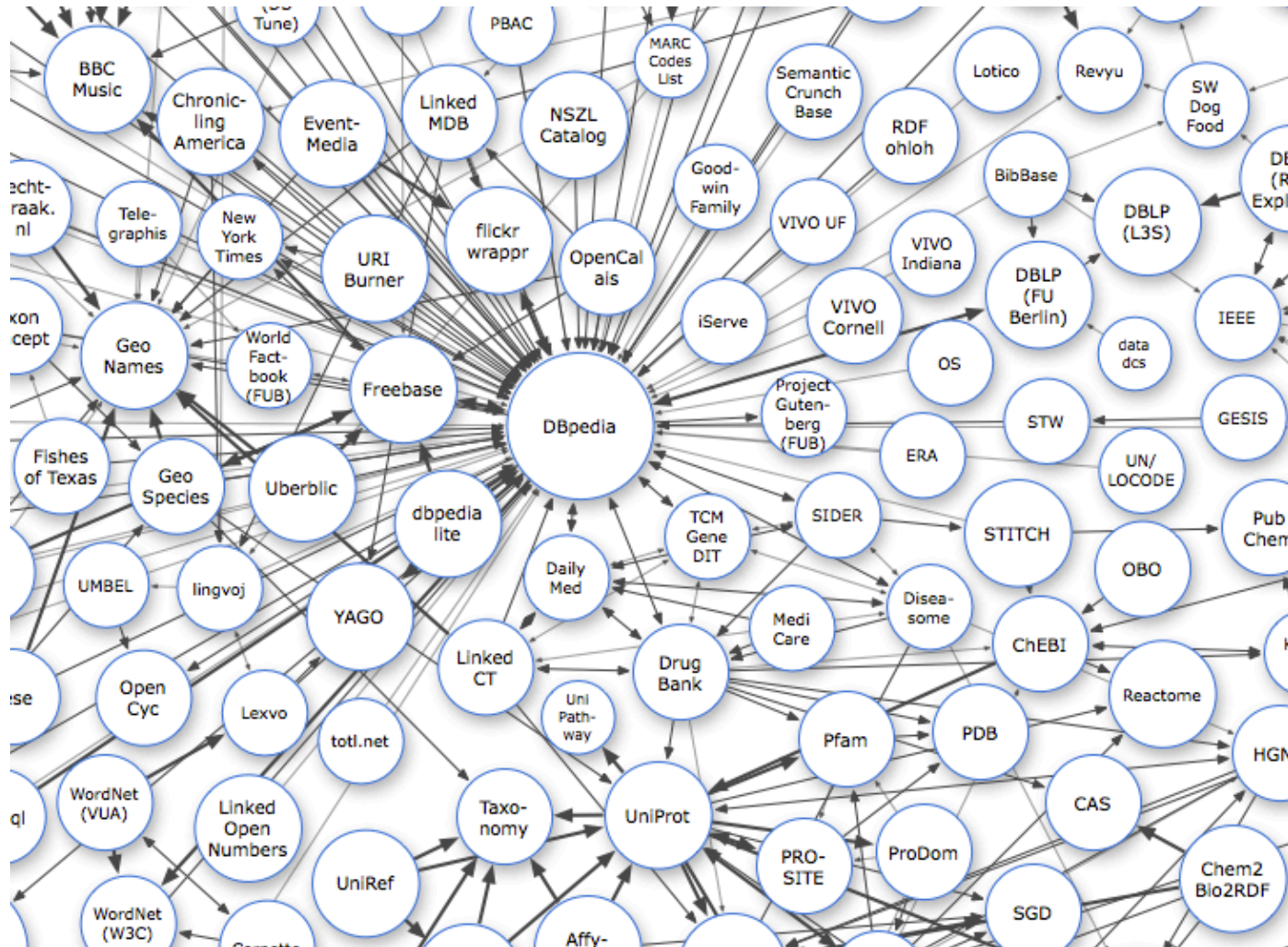
- get more information about an entity.

- use that information for coreference resolution.

- use that information for Question Answering.

The LOD Cloud

On the LOD Cloud the information comes from different ontologies



Accessing the SW through NL: problems - 1

Without NLP technique, access to the SW through Natural Language is allowed only using a short lexicon, which is made up of non homogeneous KBs labeling systems.

Homonymy: when we look up a word in DBpedia we get a long list of possible candidates. Such problems are due to both word ambiguity and to the labeling system used.

Synonymy: in DBpedia the semantic content of the synonymy is not treated, there is only a redirect system used to manage misspellings, alternative spellings, tenses, punctuation, capitalizations or to redirect sub-topic in broader context

Accessing the SW through NL: problems - 2

Some relations in DBpedia have anomalous names that are hard to understand and therefore are difficult to use.

Another problem concerns the fact that many relations share the same meaning, for example “dateOfBirth”, “birthDate” and “datebirth” are three variants of the same concept. So when we want to retrieve all the entities with a particular property we have to collect all the various forms of the property. Similarly.

DBpedia classes were extracted from different sources such as YAGO, UMBEL and Wikipedia. Many extracted classes have the same problems of properties and express complex concepts with n-ary relations, which are hard to use without a preprocessing phase.

Accessing the SW through NL: preprocessing - 1

1.CL AncientGreekPhilosophers

2.CL OlympicTennisPlayersOfTheUnitedStates

Classes of that kind have a complex semantics that is hard to use without a preprocessing phase.

We analyzed them with a syntactic constituency parser and obtained the output below, where F3 is the label for fragments, SN stands for NounPhrase, SP for PrepositionalPhrase:

1. f3-[sn-[Philosophers-n-sn, (mod)-[ancient_Greek-n-sn]]]
2. f3-[sn-[olympic-ag-sn,tennis_players-n-sn,(mod)-[of-p-sp,the-art-sn, United_States-n-sn]]]

Accessing the SW through NL: preprocessing - 2

Another step is done mapping the heads with synsets in WordNet, in order to expand the KB lexicon, for instance, the word “actress”, with our approach matches the class: dbpedia-owl:Actor, because “actress” share the same synset of “actor”, as shown in the following term,

```
dbp('Actor',[actor-n],[109765278,109767197])
```

WordNet mapping allows us to use hyponymy relation, for instance, the word “wife” matches the property: dbpedia-owl:spouse, because there is an hyponymy relation between “wife” and “spouse”.

GETARUN

The system for text understanding developed at the University of Venice, which is organized as a pipeline that includes two versions of the system: what we call the Partial and the Deep GETARUNS.

The system is based on LFG (Lexical Functional Grammar) theoretical framework and has a highly interconnected modular structure. The Closed Domain version of the system is a top-down depth-first DCG (Definite Clause Grammars) based parser written in Prolog Horn Clauses. It is divided up into a pipeline of sequential but independent modules which realize the subdivision of a parsing scheme as proposed in LFG theory where a c-structure is built before the f-structure can be projected by unification into a DAG (Direct Acyclic Graph).

Accessing the SW through NL: technique - 1

An useful ontology must support a systematic mapping to and from natural languages, because such a bridge could break the static nature of a KB and m.

What we have in mind is the assumption that an ontology reflects the background knowledge used in writing, reading and thinking ake it flexible.

We consider the computation of meaning as a set of rules that determine the reference of words. We consider common names as classes, determiners as restrictions on classes, entities as referents and verbs as relations between entities and classes. This scheme is compatible with the RDF structure and can also serve as a bridge between natural language and KBs.

Accessing the SW through NL: technique - 2

We can use names as labels to access a KB in order to get all the information regarding an entities. But as we noted the same label could refer to several entities. The solution is to use contextual information. For instance, in the following example taken from the RTE5:

(CNN) -- Malawians are rallying behind Madonna as she awaits a ruling Friday on whether she can adopt a girl from the southern African nation. The pop star, who has three children, adopted a son from Malawi in 2006. [...]

when we find an ambiguous entity (the pop start) we look for information that could disambiguate it. In this case, the singular definite expression “the pop star” is used to specify the entity Madonna.

QA over Linked Data: Logical Forms

LFs produced by GETARUNS are all expressed as complex Prolog terms, and can be decomposed into three subparts: there is a Pred - the main verb predicate of the question -, a Focus - this is the question head expressed in the question which may correspond to an interrogative pronouns or may have a nominal head -, and then Arguments - this slot contains argument head and its internal modifiers and attributes like Quantifier, Cardinality, Plural.

Which are the presidents of the United States of America?

Pred: [be]

Focus: [person],

Arg: [president/theme_bound-[['United_States_of_America']]]

QA over Linked Data: SPARQL Queries 1

The LF is used to build the SPARQL query.

Predicate [be] can be regarded as the fact "belonging to a class".

Focus [person] tells us that the reply foreseen by the question must be of Type Person, important feature which is easily expressed in SPARQL.

We then look for the elements in Arg inside the two ontologies, that we mapped with WordNet, and we obtain the class:

yago:PresidentOfTheUnitedStates.

At this point, we can start building the query according to the schema,

[?x a [Focus] . ?x a [Class]].

QA over Linked Data: SPARQL Queries 2

```
select distinct ?x ?string WHERE{  
    {?x a dbpedia-owl:Person .  
     ?x a yago:PresidentsOfTheUnitedStates}
```

```
union
```

```
    {?x a foaf:person .  
     ?x a yago:PresidentsOfTheUnitedStates}
```

```
union
```

```
    {?x a yago:Person100007846 .  
     ?x a yago:PresidentsOfTheUnitedStates}
```

```
OPTIONAL {
```

```
    ?x rdfs:label ?string .
```

```
    FILTER (lang(?string) = "en")}}
```

QA over Linked Data: SPARQL Queries 3

YES/NO QUESTIONS.

In case the LF does not produce a Focus element, the system understands that the question type is yes/no. In this case, the system will create a query of type ASK,

Is Christian Bale starring in Batman Begins?

Pred: [be], Focus: []

Arg:['Christian_Bale'/theme_bound-[mod-[pred-[star],
['Begins'/theme-[mod-[pred-['Batman']]]]]]]

Schema: [Ent Prop Obj]

QA over Linked Data: SPARQL Queries 4

In other questions we proceed by disambiguating a property contained in the LF before proceeding to build the corresponding query.

Who was the wife of President Lincoln?

Pred: [be], Focus: [person]

Arg: [wife/theme_bound-[['Lincoln'/theme-[(mod)-[pred-['President']]]]]]

The class President helps us to disambiguate the entity expressed by the question, because it contextualizes the reference, and it allows us to recover the actual intended entity, i.e.

Abraham_Lincoln. In DBpedia there are several entities with the label Lincoln, that refers to musicians and places

QA over Linked Data: Problems

In our system the major problems we had have been with the ability to recover complex concepts, as for instance in the question:

Give me all female German chancellors!

where we try to decompose the meaning into three different but intertwined queries (?x Female. ?x German. ?x Chancellor). But we don't get desired results and the reason is that DBPedia does not contain the male/female distinction.

Probably there are amalgams which can express the complex concept to be a woman and be the head of the German government, but at the moment, our mapping strategy has not been able to find a class for the concept.

QA over Linked Data: Evaluation

We have tested our system on the training set made available by QALD-11 workshop organizers. The training set contains 50 question expressed in natural language to submit to DBPedia.

We obtained correct answers (Precision and Recall =

1) to 23 questions over 50, with a final overall Precision and Recall equal to 0.46.

- a. in 14 cases, we did build up an efficient and adequate query;
- b. in 5 cases we obtains partial results F-Measure ranging 0.40-0.80
- b. in 4 cases we got a Precision ranging 0.80-0.98;
- c. in 5 cases we got a Recall ranging 0.85-0.99.

Thank you for your attention!