Question Answering Biographic Information and Social Network Powered by the Semantic Web

Peter Adolphs, Xiwen Cheng, Tina Klüwer, Hans Uszkoreit, Feiyu Xu

German Research Center for Artificial Intelligence (DFKI)
Project Office Berlin
Alt-Moabit 91c
10559 Berlin
Germany
{peter.adolphs,xiwen.cheng,kluewer,uszkoreit,feiyu}@dfki.de

Abstract

After several years of development, the vision of the Semantic Web is gradually becoming reality. Large data repositories have been created and offer semantic information in a machine-processable form for various domains. Semantic Web data can be published on the Web, gathered automatically, and reasoned about. All these developments open interesting perspectives for building a new class of domain-specific, broad-coverage information systems that overcome a long-standing bottleneck of AI systems, the notoriously incomplete knowledge base. We present a system that shows how the wealth of information in the Semantic Web can be interfaced with humans once again, using natural language for querying and answering rather than technical formalisms. Whereas current Question Answering systems typically select snippets from Web documents retrieved by a search engine, we utilize Semantic Web data, which allows us to provide natural-language answers that are tailored to the current dialog context. Furthermore, we show how to use natural language processing technologies to acquire new data and enrich existing data in a Semantic Web framework. Our system has acquired a rich biographic data resource by combining existing Semantic Web resources, which are discovered from semi-structured textual data in Web pages, with information extracted from free natural language texts.

1. Introduction

After several years of development, the vision of the Semantic Web (Berners-Lee et al., 2001) is gradually becoming reality. The technologies have matured to a point where they can be applied in production environments. At the same time more and more meaningful content is becoming available in machine-processable form and is utilized in real-world applications. Semantic Web data can be published on the Web, gathered automatically, and reasoned about. All these developments have paved the way for interesting applications of natural language technologies that interface human language with Semantic Web data.

The relation between natural language processing (NLP) and the Semantic Web is two-fold. On the one hand, NLP systems can be applied to extract structured information from free natural language texts, which in turn can be made available in a Semantic Web data format such as RDF and stored in a Semantic Web knowledge base (Buitelaar and Declerck, 2003). On the other hand, Semantic Web data can be utilized as knowledge resources for building NLP applications featuring information retrieval, semantic inference or question answering (Lopez et al., 2007).

In this paper, we present an information system for a specific domain – biographical data of famous people – that applies NLP in both outlined ways. Our system has acquired a rich biographic data resource by combining existing Semantic Web resources, which are discovered from semi-structured textual data in the Web pages, with information extracted from free natural language texts. On top of this knowledge base, we built a question answering system, which allows users to access information in a smooth natural language dialog. This system thus demonstrates how NLP methods and Semantic Web technology and resources can be fruitfully combined to build a new class of domain-specific, broad-coverage information systems.

The paper is structured as follows: section 2 motivates why Semantic Web data offer interesting opportunities for knowledge-driven artificial intelligence (AI) applications. The architecture of our own system is described in section 3. The common data model that is used for representing knowledge in the knowledge base as well as in the question processing component is described in section 4. Section 5 provides details about the different data sources that we used in our application, how they were acquired and how we merged them. Sections 6 to 8 describe the consecutive steps of the actual question answering system, namely input analysis and intepretation (section 6), answer retrieval (section 7) and multimodal answer generation (section 8). Section 9 concludes the paper and gives an outlook to future directions.

2. Semantic Web

When the vision of the Semantic Web was formed more than a decade ago, it was propagated that in the near future large parts of the existing Web pages would be annotated with logical representations, bringing “structure to the meaningful content of Web pages” (Berners-Lee et al., 2001). Furthermore, formal domain descriptions and inference rules would be made available in order to allow machines to read information from documents in a structured way, to infer knowledge, and finally to help users find and organize information much more easily and efficiently by automatically exploiting links between independently discovered information sources. Since then, technologies and formalisms for realizing the Semantic Web (RDF, RDFS, OWL, triple stores, SPARQL) have been developed and put into use.

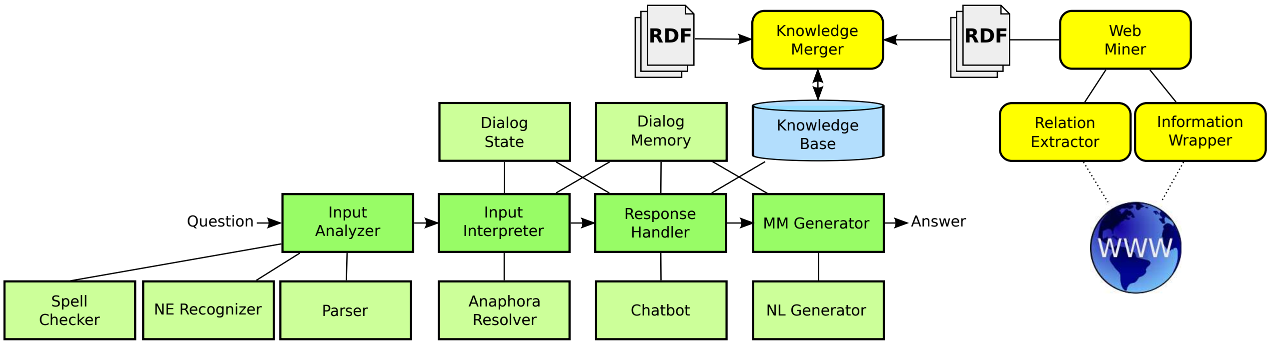

Figure 1: System components: dialog processing (angular boxes) and knowledge management (rounded boxes)

Today, most Web documents are still not accompanied by Semantic Web resources that represent the same information in a machine-readable form, as had been envisioned originally. Nevertheless, over the past years, many data repositories have been created and made publicly available using Semantic Web technology. These data were not always created with the Semantic Web in mind, but are derived representations of existing structured resources, such as the digital bibliography DBLP, the audio database MusicBrainz or the lexical database WordNet, to name just a few. Other resources contain information extracted from semi-structured parts of Web sites, such as the World Factbook or Wikipedia. Most of these data make assertions about (specific aspects of) the world, rather than just Web documents, which make them valuable resources for knowledge-driven applications in Artificial Intelligence (AI). By linking data from different origins[1], some of which are community-driven, an even larger and richer knowledge resource is taking shape, potentially widening a long-standing bottleneck of AI systems, the notoriously incomplete knowledge base.

3. System Architecture

As depicted in figure 1, our system comprises two subsystems: an online part (green-colored, with angular boxes) and an offline part (yellow-colored, with rounded boxes). The online subsystem is the dialog-enabled question answering (QA) component, while the offline part is responsible for the acquisition and management of knowledge.

The question answering part of our system is realized in a classical pipeline architecture: When a user poses a query, it is first linguistically analyzed by the *Input Analyzer*, then interpreted with respect to the current dialog context by the *Input Interpreter*. The result of the input interpretation is a semantic representation of the user's question. A *Response Handler* turns the question semantics into a query to the *Knowledge Base*, and passes the answer, in an abstract representation, to the *Multimodal Generator*, which finally generates and presents the answer to the user in multiple ways.

As detailed elsewhere (Xu et al., 2009), our system is able to provide the QA functionality in a smooth and connected dialog. The models of the *Dialog Memory* and the *Dialog State* allow us to resolve pronouns from previous entity mentions as well as to pose and react to clarification questions in case of a possible misunderstanding. A *Chatbot* component helps us to reply to a user's utterance when the system cannot arrive at an interpretation.

The *Knowledge Base* (KB) is of central importance to our system, since it provides knowledge for the question answering and stores the validated data resulting from information wrapping, information extraction and information merging.

4. RDF as a Unified Semantic Model

The Resource Description Framework (RDF) is one of the cornerstones of the Semantic Web. The term is used to refer both to the underlying data model of the Semantic Web as well as to a specific XML-based exchange format (which is of no interest here). The main advantage of the data model is its simplicity. An RDF statement merely consists of a binary relation between two entities, that is a triple of a predicate and two arguments – the 'subject' and the 'object' – which are conventionally written in the order *subject predicate object*. Entities and predicates are referred to with URIs (leaving primitive types such as strings and numbers aside for now). This guarantees that the vocabulary defined for a specific resource has a unique namespace, which makes it easy to mix and combine vocabularies and data of different origin in the same KB.

All data in our KB as well as the semantic representations of natural language questions and answers are represented in an RDF form. Figure 2 shows some example statements from the KB that express the fact that there is a person, identified by `g:Person.14193`, which is called "Madonna", as well as another person, identified by `g:Person.119944`, which is called "Carlos Leon", and that there is a "hasBoyfriend" relation, identified by `g:hasBoyfriend` between the former and the latter. As said before, identifiers such as `g:Person.14193`, `g:hasBoyfriend` or `rdf:type` are all URIs, here in a prefixed form where the prefix `rdf:` represents the namespace for the basic RDF vocabulary and `g:` represents the namespace of our gossip data.

Our gossip data is additionally structured by an ontology, specified in the Web Ontology Language (OWL). The ontology serves as the domain model, which defines classes of instances, sub-class hierarchies and properties of these classes. The strength of the ontology is unleashed when

[1] see also the Linking Open Data community project, `http://linkeddata.org/`, accessed on 17 Mar 2010

Subject	Predicate	Object
g:Person.14193	rdf:type	g:Person
g:Person.119944	rdf:type	g:Person
g:Person.14193	g:hasName	"Madonna"
g:Person.119944	g:hasName	"Carlos Leon"
g:Person.14193	g:hasBoyfriend	g:Person.119944

Figure 2: Example statements in the KB

it is combined with an inference engine, which evaluates the domain definitions and makes implicit knowledge explicit by asserting triples that must also hold true for generalizations made in the ontology. For example, the property `g:hasWife` has the domain `g:Woman` according to the ontology, which in turn is defined as having the property `g:hasGender` with the value "female", allowing the inferencing engine to assert the "female" property for every wife in the KB. Likewise, we employ a taxonomy for relations between people which allows us to infer less specific relations between people if a more specific relation is explicitely stated. For instance, if X has a husband Y, then X also has a spouse Y, a partner Y and – more general – a relationship to Y and a connection to Y. Thus, by modelling the personal relation taxonomy in the ontology, we make implicit knowledge explicit in an elegant and declaritive way, rather than creating redundant data from the acquired information with ad-hoc code or building a non-redundant data store which has to be accessed with more complex queries.

We use SwiftOwlim[2] for storing and querying the RDF data. SwiftOwlim is a 'triple store', a storage component that is specifically tailored toward Semantic Web data. It provides a forward chaining inference engine, supporting a subset of OWL Full. Once the reasoner is finished, the triple store can be queried directly using the RDF query language SPARQL or it can be dumped into a relational database and then queried with SQL.

5. Knowledge Acquisition and Merging

Our Knowledge Base (KB) is populated with data from different sources. One of the data resources we acquired ourselves, whereas the other stems from an existing Semantic Web resource. Although all data is modelled in RDF, the richness of the KB only becomes apparent when the data share a common vocabulary whereever they refer to the same things. Therefore a common OWL ontology serves as a domain model in that it defines the common vocabulary and is used for inferring implicit knowledge from data of potentially different origins (see section 4.). If the data sources (partially) cover the same domain, as it is the case here, duplicated instances in the two data sources have to be found and merged.

In the first version of our system (Xu et al., 2009), we created a database with bibliographical information about people in the pop music domain, the RASCALLI KB. It was extracted from the Web by two different means: i) using Information Wrapping techniques for extracting pieces of information from structured and semi-structured parts of Web sites, and ii) employing a minimally supervised machine learning method for acquiring relation instances from free text using the DARE system (Xu et al., 2007). DARE automatically learns linguistic patterns indicating a mentioning of a target semantic relation, starting with only some trusted relation instances as the initial seeds in a bootstrapping learning process. Once the patterns for a specific semantic relation have been learned, they can be used on free text to find previously unknown relation instances.

In the present system, the pop musician data was supplemented with the biographical data on famous people of all domains from YAGO (Suchanek et al., 2007). YAGO is a huge ontology which has been automatically extracted from the semi-structured parts of Wikipedia and the taxonomic structure of WordNet. Each Wikipedia page is considered a candidate for becoming an instance in YAGO. If the page has an infobox, it is parsed and the relevant properties of the candidate instance are extracted by a set of heuristics. The page category system of Wikipedia is further exploited by another set of heuristics in order to guess the class of that candidate instance.

When combining two different knowledge sources with overlapping domains, it is crucial to solve the object identity resolution task (Elmagarmid et al., 2007). This involves finding records in both data sources that refer to the same entities in the world. The RASCALLI KB and YAGO have overlapping concepts and instances for people, groups and locations. As to the conceptual model, we manually extended the RASCALLI ontology by adding missing properties such as `y:actedIn` or `y:wrote` from the corrosponding concepts in YAGO. The instances were merged by applying a set of heuristics based on common sense as well as on cultural-specific knowledge. For example, we assumed that two instances of the type *person* belong to the same entity when they share at least some features such as their name descriptions, birthdays or birthplaces. However, it occurs in many cases that some important features are unavailable for our acquired instances or that their property values do not match exactly to each other. In such cases, we decided to soften the match constraints. For instance, we have to utilize a normalization rewriting grammar for names in order to cope with spelling variants or we have to make use of the existence of common relatives.

The resulting KB contains information about 734,865 individuals, for instance:

- 618,445 persons
- 50,601 published material
- 34,458 movies
- 20,733 locations

The asserted relation instances contain the following, among others:

- 734,865 "has name"
- 442,319 "born on date"
- 205,808 "died on date"
- 18,793 "has partner"
- 12,594 "has parent"

In total, 5,081,456 triples are asserted.

[2] `http://www.ontotext.com/owlim/`, as accessed on 17 Mar 2010

6. Robust Semantic Question Processing

In our system, user utterances are first annotated with domain-independent linguistic analyses, and are then assigned domain- and context-specific interpretations. The *Input Analyzer* is responsible for deriving a propositional meaning from the user's utterance by taking the lexical semantics of the words and the syntactic analysis of the sentence as input. It applies several off-the-shelf linguistic tools, namely a spell checker, a named entity recognizer (Drozdzynski et al., 2004), and a dependency parser (de Marneffe and Manning, 2008). The *Input Interpreter*, on the other hand, acts at a domain-specific level by assigning a meaning in the context of the conversation. Furthermore, pronoun resolution takes place at this stage as well as the mapping from linguistic predicates to domain-specific predicates which are used in the KB.

In order to achieve robustness and accuracy for question processing, we take a hybrid approach by combining two strategies. One is a fuzzy pattern matching algorithm which utilizes regular lexico-syntactic patterns based on surface strings and recognized named entities, while another makes use of the dependency tree structures as patterns. The lexico-syntactic patterns are very robust and not dependent on the performance of a parser. However, the enhancement of the pattern set with the dependency trees allows us to reduce the 1067 lexico-syntactic patterns to 212 dependency tree patterns, with almost the same linguistic coverage. Thus the utilization of syntactic parsing results eases maintenance of the pattern set in a significant way (see also (Klüwer, 2009)).

The semantic representation of questions consists of information about the propositional content, the question focus and the question type, as shown in (1). Propositional content is coded in a predicate-argument structure with a predicate (e.g., "hasBoyfriend") and two arguments (e.g., "Person.14193" and the query variable "?"). All elements in the predicate argument structures can be either instantiated with some values or are to be filled with the answer values, acting as the question foci.

(1) "Who is the boyfriend of Madonna?"
⟨hasBoyfriend, Person.14193 ?⟩, [wh]
⟨*RELATION, ARG1 ARG2*⟩, [*question-type*]

By mapping utterances of quite different sentence types but the same dialog function such as plain questions ("Who is Madonna?"), statements with embedded questions ("I wonder who Madonna is."), statements about the user's interests without embedded questions ("I'm interested in Madonna.") to the same semantic representation, we can moreover conflate sets of user utterances with the same intended meaning and are thus able to cover a wider variety of input.

7. RDF-based Answer Retrieval

Answer retrieval is realized by i) mapping the semantic representation of a user's question to the KB query language, that is SPARQL or SQL, depending on the specific data store, and ii) executing the query in order to get a set of fillers for the queried variable(s) as the underlying logical answer to the user's question. Owing to the simplicity of the RDF data model and our semantic representation reflecting that model, it is sufficient to determine the query by choosing the right query template based on the blank variable(s) and instantiating it with the specific values at hand. For instance, the semantic representation in (1) is realized as the SPARQL query in (2).

(2)
```
SELECT $x { g:Person.14193 g:hasBoyfriend $x }
```

The advantage of our RDF-based question semantics is that it conflates similar information requests that differ only in more subtle aspects of language use such as politeness. Furthermore, questions involving the same individuals and properties but differing in their expected answer types (EATs) such as person ("who?"), quantity ("how many?") or truth value (yes/no question) share the same question triple and differ only in their question type value. This allows us to map all factoid questions that our system can deal with to fewer than the 8 theoretically possible query patterns for finding an answer to a single question triple and deal with the different EATs in a later step. For instance, the questions "Who are the boyfriends of Madonna?", "Does Madonna have any boyfriends?" and "How many boyfriends does Madonna have?" are all mapped the SPARQL query in (2).

8. Multimodal Answer Generation

In order to increase the naturalness of the dialog between user and system, we output the answer couched in natural language, rather than realizing it as a list of facts or, even worse, a list of RDF resource URIs. We employ a template-based generator for realizing the logical answer verbally, controlled only by a small set of parameters. Depending on the question semantics, which forms the topic expression of the verbal answer to be produced, and here in particular the queried predicate and the presence of the arguments, the EAT and the answer size, an appropriate answer template is selected, as illustrated in table 1. The same parameters that are used to condition the template selection can also be used to instantiate the slots in the verbal answer pattern, once it was chosen. The answer size parameter, for instance, will also be chosen as a slot filler for answers with the EAT 'quantity'. Relativizing expressions such as "as far as I know" or "to my knowledge" are inserted at times to signal to the user that the knowledge base might not be complete. Furthermore, we usually provide several natural language answer templates for the same logical answer, among which one is selected randomly, in order to create a more vivid dialog.

Some answers are too complex to be realized only verbally. For instance, answers concerning a person's connections to other people would be very dull if they were carried out by just listing the names and relations of others to that person. Answers to such questions are better understood if they are given using different modalities. What is more, Semantic Web data usually makes rich use of links to human-readable Web resources, which would be lost if we could not easily integrate them into the answer. Therefore, we made it possible to give or enhance an answer with multimedial content in a Web browser (Xu et al., 2009), choosing the ideal

Semantic Predicate	**EAT**	**Answer Size**	**Answer Strategy**
`g:hasBoyfriend`	Person	≥ 1	Output answer from KB (list of people)
`g:hasBoyfriend`	Quantity	≥ 1	Return size of answer triples
`g:hasBoyfriend`	Truth Value	≥ 1	Return “yes” and support answer with some examples
`g:hasBoyfriend`	Any	$= 0$	Return “I don’t know” + supportive answer (Google search)
`g:hasDeathDay`	Time	$= 1$	Output answer from KB
`g:hasDeathDay`	Time	> 1	Output answer from KB; hint at contradictionary status of answer
`g:hasDeathDay`	Time	$= 0$	Return “x is still alive” + supportive answer (Google search)
Any	Any	$= 0$	Return “Sorry, I don’t have that information”

Table 1: Template selection for natural language answer generation

representation for each answer. Verbal answers can be supported by visualizations of social network graphs, connection paths, location maps, personal profile and homepages, or by the IMDB pages for a movie. Further visualization types for other types of answers can easily be added.

9. Conclusion

In this paper, we have presented a system that shows how the wealth of information in the Semantic Web can be interfaced with humans once again, using natural language for querying and answering rather than technical formalisms. Furthermore, we have shown how to acquire new data and enrich existing data using NLP technologies in a Semantic Web framework. Of course, there is already much more data available than what we have used here, and we plan to integrate more of these resources into our system. Considering the domain of our application, social media that have arisen in the Web 2.0 context are of a special interest to us. In cases where these media are not available in a semantically processed format, we plan to further refine our information extraction methods using these data in order to create new resources that will feed the knowledge base of our application.

10. Acknowledgements

The work presented here was partially supported by the international project RASCALLI, funded by the Sixth Framework Programme of the European Commission in the area of Cognitive Systems (IST-27596-2004), by a grant to the project KomParse by the ProFIT programme of the Federal State of Berlin, co-financed by the European Union through the EFRE fund (Europäischer Fonds für regionale Entwicklung / European Regional Development Fund), as well as through a research grant to the project TAKE, funded by the German Ministry for Education and Research (BMBF, FKZ: 01IW08003).

11. References

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *The Scientific American*, May.

Paul Buitelaar and Thierry Declerck. 2003. Linguistic annotation for the semantic web. In Siegfried Handschuh and Steffen Staab, editors, *Annotation for the Semantic Web*, volume 96 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

Marie C. de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, Manchester, UK.

Witold Drozdzynski, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. 2004. Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz*, 1:17–23.

Ahmed Elmagarmid, Ipeirotis Panagiotis, and Vassilios Verykios. 2007. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(1), January.

T. Klüwer. 2009. RMRSBot – using linguistic information to enrich a chatbot. In *Proceedings of IVA 2009)*, Amsterdam, The Netherlands.

Vanessa Lopez, Victoria Uren, Enrico Motta, and Michele Pasin. 2007. Aqualog: An ontology-driven question answering system for organizational semantic intranets. *Journal of Web Semantics*, 5(2):72–105.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of WWW 2007*, New York, NY, USA.

Feiyu Xu, Hans Uszkoreit, and Hong Li. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. *Proceedings of ACL-2007.*

Feiyu Xu, Peter Adolphs, Hans Uszkoreit, Xiwen Cheng, and Hong Li. 2009. Gossip galore: A conversational web agent for collecting and sharing pop trivia. In *Proceedings of ICAART 2009*, Porto, Portugal.