Asking the Right Question: How to Transform Multilingual Unstructured Data to Query Semantic Databases

Andrés Domínguez Burgos, Koen Kerremans, and Rita Temmerman

Centre for Special Language Studies and Communication
Department of Applied Linguistics
Erasmus University College Brussels
Pleinlaan 5, B-1050 Brussels, Belgium
{andres.dominguez.burgos,koen.kerremans,rita.temmerman}@ehb.be

Abstract. Ontology engineers have long tried to develop mechanisms to automatically transform natural language statements into queries knowledge-systems can deal with. This has been an enormous challenge as natural languages are highly ambiguous and contexts for disambiguating are seldom identifiable through simple linguistic patterns. To circumvent these difficulties, developers of knowledge bases have often opted for the use of a restricted vocabulary and syntax. Normal users, nevertheless, prefer to express themselves in their language. Special languages or schemas tend to reflect one language – the developer's – and make extensibility more difficult. Also multilingual access can be more difficult to handle in that way. In this article we present strategies for transforming queries of natural languages into language-neutral representations that can be more easily transformed into semantic queries. We describe a tool that combines a multilingual database and natural processing modules with a semantic database in order to transform queries in Dutch, French and English into queries from which ambiguity at syntactic and semantic levels have been reduced. We focus on certain aspects of natural language such as negation and collocation preferences to deal with semantics.

Keywords: collocation, natural language queries, negation, structured data, ontology, unstructured data.

1 Introduction

The Semantic Web was conceived with the vision of making the access to knowledge more accessible through the use of semantic information. This semantics was conceived mainly as semantic markers that people or automatic processes add to structured and unstructured data. Humans, though, prefer to use natural language. If they want to search these semantic web repositories, they need to 1) try different keywords or 2) interact by using some query language. The first approach – semantic marking by humans - is time-consuming and ultimately frustrating for users. If users restrict themselves to key words, they cannot resort to constraints that help making the search

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 262–271, 2012.

more precise. Using a query language is not an option for most users – it implies learning non-trivial syntax, getting to know a certain vocabulary and having an idea about the structure of the data model behind the data collection the user wants to search. The need to facilitate semantic search has a long tradition. Seminal work can be traced back to the first efforts to create interfaces to access conventional databases in the late sixties. Enhancing search through the use of semantics has been one of the key research areas in the Web in the last few years [1]. In section 2, we present the state of the art. Section 3 discusses general issues regarding ambiguity in natural language queries. Section 4 shows an overview of the natural language components of LinSigna, a tool that converts natural language queries from different languages into a normalized form, a semantic representation, used for querying. Section 5 discusses initial results. Section 6 reports on work ahead and section 7 has the conclusions.

2 State of the Art

The transformation of natural language into some formal representation has a long tradition in different fields: from machine translation and Q&A systems to semantic search. [2] presented an overview of problems found when trying to close the breach between natural language and databases and discussed some of the initial work.

Querix was a natural language interface to the Sematic Web [3]. The system does not try to solve human-language ambiguities but asks questions to the user for further clarifications. Ginseng was another tool that presented itself as a semantic engine with a full natural language interface. In reality, the system used an entry form that shaped – or restricted – the kind of queries that can be asked. Keywords are "suggested" from the vocabulary loaded from the chosen ontologies [4].

Form-based interfaces that guide users can be well suited for domain-specific purposes but they are difficult to extent. As Damjanovic points out, the development of natural language interfaces (NLI) has not been very widespread because of the high development and customization costs associated with them. SemSearch and AquaLog [5] were some other attempts. The former did not deal with natural language processing. A google-like query interface allowed users to specify the subject and three basic operators: ":" for specifying and the boolean operators "and" and "or". It matched keys to the labels of concepts and semantic relationships. Finally, it used query templates to transform user queries into formal queries. AquaLog used WordNet and shallow parsing. One of the problems was that it required syntactically correct sentences. AutoSparql [6] was another attempt to enable queries in natural language and converting them into SPARQL. QuestIO was a tool that interacted with ontologies using unconstrained language-based queries [7]. It aimed at minimal customization and robustness. [8] describes an approach for dynamically extending the vocabulary that supports a controlled natural language.

Summing it up, the main advantages for a natural language component are: a) external users do not need to learn a query language or have an understanding of the data model, b) users only need to use an input field and c) some user errors' are

usually dealt with natural language interface for data bases (NLIDBs) – at least to some extent. The main disadvantages are as follows: a) a system for natural language processing can only cover a subset of possible human inputs – those it has been trained to cover; humans tend to under-estimate human language productivity and complexity and b) it is difficult for the system to give an appropriate feedback when an input was not understood. We believe a NLIDB interface offers other major advantages if rightly conceived: it can help in keeping data models and general semantic mechanisms apart from human language issues and it becomes easier to add other human languages to the interface. In this article, we present some mechanisms we have developed to allow more user-friendly queries.

3 Some Issues in Interpreting Natural Language for Queries

Our general aim was to develop a system that could interpret short but non-trivial queries in different languages without naïve users having to learn some restricted language and use this interpretation to produce semantic queries. We believe the idea should be to make users go beyond isolated key word queries without having them to formulate queries in full-blown sentences, for which many do not have the time.

There has been a discussion on how users modify across time their own strategies for search [9, 10]. These strategies are the product of trial and error with the systems available in document search, in QA systems or in systems that try to take a hybrid attempt. Organizations working on search technology keep adjusting their strategies to cope with users' queries. Users and engines have been adapting their strategies over the years responding to societal and technological evolution. We think a more robust interface will enable users to ask more complex queries.

The current situation can be described as follows: many users do not want to deal or do not even know how to deal with formalisms for logic operators. Still, they express those operators routinely in their own natural language. Users are less inclined to type in long sentences reflecting some formal language syntax. Still, they can intuitively formulate queries in natural language that enables us to produce a unified representation of a complex query. This formal representation can be language-independent and can be more easily translated into a SPARQL query or another database query language.

The system we have developed reads short queries in Dutch, English or French and transforms them into a formal language Common Semantics (CS) that can be easily parsed to produce semantic queries. First, the system does a shallow parsing of the sentences. After that, the system identifies the natural language elements that may stand for logic operators. It then identifies scope and precedence of the operators. After that, it detects the possible candidates for entities, concepts and relationships. Once this process is concluded, the system tries to solve some conceptual disambiguation when it has encountered several candidates for concepts or relationships. We understand for a concept a unit of meaning that can be located in a hierarchy with other objects and which contains specific properties.

3.1 Conceptual Ambiguity

Conceptual ambiguity is a well-known phenomenon, which has not yet been entirely solved. Natural language terms are highly ambiguous. Even in domain specific utterances, a given term can often be employed to refer to different concepts.

3.2 Syntactic Ambiguity and Scope

Negation

Negation in natural language can be applied to alter the meaning of nouns, verbs, adjectives, whole nominal or verbal phrases. Negation can also have different scopes. Although the mechanisms for expressing negations have been extensively studied, few work has been carried out on the automatic detection of negation semantics. [11] works on strategies for detecting negation by identifying the focal term through a learning algorithm. The following example is a Dutch query in which a user expresses that (s)he wants to see a movie, provided that it is not an action movie or a movie for children:

>> *Film, Actiefilm noch kinderfilm*

In English, this Dutch query can be literally translated as follows: "Movie, neither action nor children's movie". The pattern "neither…nor …" points at another scope than "… noch …". The following English and Dutch queries express the search for any concert, except for world music concerts:

>> *Concert anything but world music*

>> *Concert geen wereldmuziek*

To be able to analyse these patterns, we developed a simple rule-based module called ScopeIdentifier (see section 4). Our system is designed to capture negations and resolve their scope for each language. The negation can be transformed into a negation of a concept, of a table or other collection of semantic items.

Modifiers

A typical problem in natural language processing is to ascertain the real scope of prepositions and other items with the role of modifiers. This happens very often in the case of prepositions linking two nouns or noun phrases. In the query "I want to go to the movies on Saturday", the pattern 'on Saturday' primarily refers to the predicate. In the query "Movies on science", the pattern "on science" is likely to determine and restrict the scope of movies. Different languages have different strategies to solve this and this issue again needs to be solved in the parsing mechanisms for each separate language.

All in all, different languages develop different strategies for shortening queries. Our initial observations with native speakers adapting the queries for tests has been that French speakers tend to skip certain prepositions, more so than articles. English and Dutch speakers tend to eliminate prepositions less often but virtually all articles. Linguistic rules for finding the syntactic relationship between the parts – mostly groups of nouns and how they depend on each other, where the modifier is and where the modified – obviously depend from language to language.

4 Tool Design

LinSigna is a tool that transforms natural language queries into a formal language – a common and unambiguous representation which we call CS, Common Semantics. This representation can be later used to generate semantic queries.

LinSigna uses a basic set of natural language processing modules to generate a shallow parsing of the input, identify the possible concepts and relationships, disambiguate concepts and solve the scope of logical operators based on rules. Once this is carried out, a template matrix generates the possible statements in the formal language. After that, a normalized interpretation of the input is produced.

LinSigna uses two sets of linguistic data resources: 1) a resource containing basic dictionaries and inflection forms, part-of-speech disambiguation rules and syntactic rules for the general linguistic analysis (including shallow parsing) and 2) a linguistic-semantic resource that is used for tagging the linguistic elements with conceptual items and for the process of sense disambiguation. The second resource is most relevant for this work and will be further discussed in section 4.1. In section 4.2, we will list the software components that produce the analysis and describe the general analysis of a human language query.

4.1 Linguistic-Semantic Model

The linguistic-semantic model comprises both linguistic and semantic layers. The most important items in the semantic layer are the concepts. Concepts are interlinked by means of semantic relations. They can be visualized by means of concept labels in several natural languages. These concept labels should not be confused with linguistic terms. They can be seen as base material to build up an ontology. A given concept can be connected to one or more term groups in any natural language. A term group is a set of terms that for a given context are connected to a given concept. A term X can be present in one or more term groups, which means it is connected to different concepts. The term "film" (see picture 1), for instance, can be connected to the concept of film as artistic product – a movie - or as the photographic film.

A term in a given term group can appear in collocations, i.e. linguistic patterns in which terms frequently tend to appear together. A collocation record is a lexical relation (LER) between the focal term (film) and a collocate like "shoot". An LER is similar to the lexical functions (LFs) in the Text-Meaning Theory [12]. The main difference between the LERs and the LFs is that the former can have more than one value. LERs with more than one value assign a probability for each value. For instance, given a LER x, it can have a value v1 with a probability 0.9 and a value v2 with a probability 0.1. That means that the lexical relationship is expressed 90% of the time in a given corpus with a value v1 (for instance, *watch a movie* as opposed to *see a movie*). The LERs can be used to try to determine whether an ambiguous term found in unstructured data should be better linked ton one or the other term group – depending on the values present for the LERs in those term groups in the linguistic database. Different LERs for a given focal term may signal different meanings of the term. LER values are thus context terms that can be used for disambiguation and that have been classified in our database. We will discuss this further in 4.3.

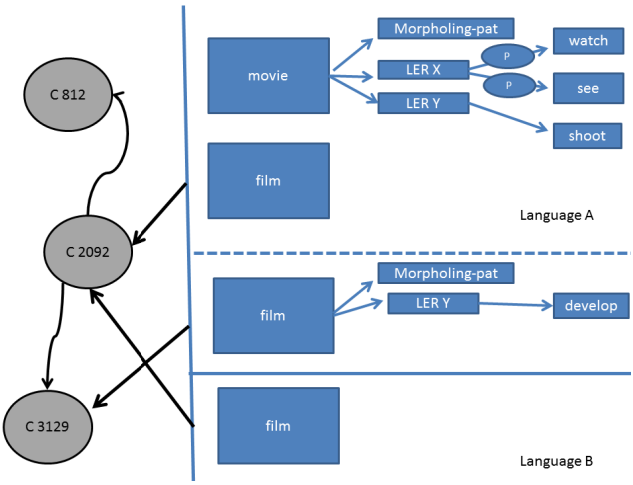

Fig. 1. The linguistic semantic model connects the linguistic layer containing possible terms in different languages (right side) with a concept in the semantic layer (left side)

4.2 Software Modules and Process

LinSigna contains the following modules: the Sentence Analyzer, the TermConcept-Mapper, the SenseSelector, the ConnectorFinder, the ScopeIdentifier and finally the StatementGenerator. Later on we plan to build a module to convert the normalized, disambiguated output into database queries.

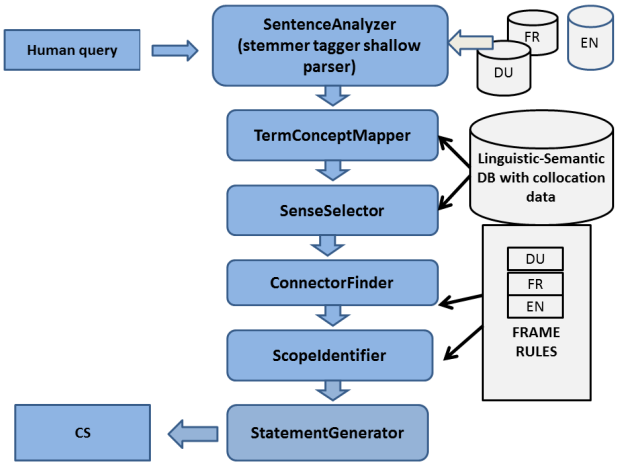

Fig. 2. LinSigna's main components

The process is illustrated in figure 2. First, a human language input *I* goes into a natural language analysis module. We developed this basic module ourselves as part of a suite of tools called Termonto Platform that helps in building up lexical and ontological data. Here, *I* passes through tokenization, part-of-speech tagging, stemming and shallow parsing. The system also tries to recover missing diacritics for Dutch or French.

Once this stage ends, the tool identifies possible candidate concepts for the input terms. Some terms may appear in several term groups and point at different concepts in the linguistic semantic resource. The disambiguation process takes two steps: syntactically-based disambiguation and context-based sense disambiguation. The SenseSelector checks for each of the concept candidates whether the collocate values match with other terms in the input query *I*. Sometimes the possible collocation values give a hint. If *I* is "what cinema shows Untouchables?", the system will identify that the term "show" is a frequent LER of the term cinema when related to the concept of movie theatre as opposed to movies in general. LER values (as seen in Fig 1) were previously obtained by processing large amounts of documents from a general and a domain specific corpora. Collocates were automatically classified by part of speech and frequency and assigned a LER label by semi-automatic means – some values tend to perform the same semantic relation but need to be verified. Some LERs have a limited set of possible values – like prepositions or verbs denoting realization. A term can have different or the same values when linked to two different concepts. Only when the values for one sense and the other are different or their likelihood for one sense is much greater than for the other can they be used for sense disambiguation. Found collocations through LERs increase the assignment of a term to a concept.

The SenseSelector proceeds to see for terms with more than one candidate concept whether those concepts are connected to other concepts linked to the other terms in *I*. If the semantic layer has a concept for the action of <show> and this is connected to the concept of <cinema as movie theater>, this latter concept will be given more weight than the concept of <cinema as art form>. Until now we have used a rather trivial way to increase the possibility that a candidate concept is the right one based on other concepts in the statement: we give a preference to the candidate with more connections in the semantic layer. In the future, we will consider the kind of semantic relationship existing between the different candidate concepts. Likewise, the tool is simply checking out whether a given ambiguous term in a human statement is present with other terms that are registered in the linguistic layer as values for one or the other term group and thus can help in assigning the right concept. In a next stage, we would like to consider reinforcement of candidate concepts that are only indirectly linked in the semantic layer. We will need to take into account two things here: how much time this connectivity checkup between candidate concepts will take and what semantic relation types are most reliable.

The ConnectorFinder proceeds to identify different operators. Depending on the natural language, these operators may appear at different positions in a sentence. The system is, for the moment, ruled-based. We believe this suffices with the kind of short queries normal users are bound to choose. In English, a user may negate with neither X nor Y. In Dutch, this pattern corresponds with the pattern 'X noch Y'. The Connec-

torFinder maps words in the input to the best logical operator. For the moment, the analysis is restricted to deal with basic patterns for basic operators as well as interrogative function words that help to identify possible tables in a database. The ScopeIdentifier tries to identify the scope of the operators (what is negated) by using predefined frames and reorders the sense if necessary. The StatementGenerator reorders the identified components according to frames.

5 Initial Results

We have decided to elaborate our own queries. The main reason for this is that we wanted to use a specific domain where multilingualism plays a very important role: Brussels and its cultural events. We also decided to elaborate queries that are natural for normal users: statements that go beyond lists of keywords but that are often shortened versions of standard sentences, i.e. statements that show already a succinct but clear structure. We derived inspiration from the data set of [13] but adapted it according to the previous criteria. Initially we have 80 queries for English and the same amount for Dutch and French. The Dutch and French queries are semantically equivalent to the English original, even if they are adapted to language traditions of shortening messages. It can be argued that the adaptations, especially the idiomatic changes, inevitably lead to differences in focus and intention, but we believe users in these three languages would basically use such statements to look for the same piece of information. Three examples (in English, French and CS) are shown in table 1.

Table 1. Examples of queries and their Common Semantic expression

English	French	CS
Rock concerts in Brussels during weekend	Concerts rock à Bruxelles pendant le weekend	concert#rock where location#Brussels, when#Monday-Sunday
Family movie after 8pm	Comédie familiale après 8pm	movie#family where time#>=20
Theatre in Ixelles in French	Théâtre à Ixelles en Français	theatre#any where location#Ixelles, language#French

So far we have managed to produce rules for interpreting basic negations, conjunctions, different forms of time and duration and location. The system manages to produce as output similar statements in a formal language (CS) whether the original input was in English, Dutch or French.

6 Work Ahead

Our next step is to implement a converter to SPARQL. Once that is done we will be able to test the system's response on a large scale to test queries in Dutch, English and French against real data. The translation of those queries into CS statements should produce basically comparable results for a large amount of cases. As mentioned in Section 3, one of the problems of natural language interfaces is how to send back in an appropriate manner a feedback when the query has not been understood or other problems occurred. Several strategies will need to be worked out for different cases: a) when the error is a product of the human language input – either because the query was not properly formulated or the system is not robust enough to analyze it, b) when there is a limitation in the expressiveness of CS and c) when the database was properly accessed but no result was found. The system currently deals with one domain – cultural events - and possible common – ambiguous – general language that can appear in queries related to it, but we believe the way it is conceived allows for expansion to other domains. We think it offers more power than one that deals with isolated key words only or one that forces users to write full formal sentences and special languages.

7 Conclusion

In this article, we presented the LinSigna tool combining a multilingual database and natural processing modules with a semantic database in order to transform natural language queries into semantic queries from which ambiguity at syntactic and semantic levels have been reduced. The tool we have described so far and the strategies behind it can enable semantic search for naïve users using different natural languages. They won't have to restrict themselves to a form of pseudo English or fill in cumbersome forms but state a query in their own language.

Acknowledgments. This research has been partially financed with funds from the Brussels Region in the framework of the Open Semantic Cloud for Brussels project in the Innoviris program.

References

1. Damjanovic, V., Kurz, T., Westenthaler, R., Behrendt, W., Gruber, A., Schaffert, S.: Semantic Enhancement: The Key to Massive and Heterogeneous Data Pools. In: Proceedings of IEEE-ERK, Portoroz (2011)
2. Chandra, Y.: Natural language interfaces to databases (2006)
3. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 980–981. Springer, Heidelberg (2006)

4. Bernstein, A., Kaufmann, E., Kaiser, C., Kiefer, C.: Ginseng: A guided input natural language search engine for querying ontologies. In: Jena User Conference, pp. 144–157 (2006)
5. Lopez, V., Pasin, M., Motta, E.: AquaLog: An Ontology-Portable Question Answering System for the Semantic Web. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 546–562. Springer, Heidelberg (2005)
6. Lehmann, J., Bühmann, L.: AutoSPARQL: Let Users Query Your Knowledge Base. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 63–79. Springer, Heidelberg (2011)
7. Damljanovic, D., Tablan, V., Bontcheva, K.: A text-based query interface to owl ontologies. In: 6th Language Resources and Evaluation Conference (LREC), Marrakech, Morocco (2008)
8. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. The Semantic Web: Research and Applications, 106–120 (2010)
9. Muramatsu, J., Pratt, W.: Transparent Queries: investigation users' mental models of search engines. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 217–224 (2001)
10. Huang, J., Efthimiadis, E.N.: Analyzing and evaluating query reformulation strategies in web search logs. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 77–86 (2009)
11. Blanco, E., Moldovan, D.: Semantic representation of negation using focus detection. In: Proceedings of 49th Annual Meeting of the Association for Computational Linguistics, pp. 19–24 (2011)
12. Mel'čuk, I.: Lexical functions: a tool for the description of lexical relations in a lexicon. Lexical Functions in Lexicography and Natural Language Processing 31, 37–102 (1996)
13. Quarteroni, S., Guerrisi, V., La Torre, P.: Evaluating Multi-focus Natural Language Queries over Data Services. In: Proceedings of LREC 2012, Istanbul, pp. 2547–2552 (2012)