Semantic Processing of Natural Language Queries in the OntoNL Framework

Anastasia Karanastasi and Stavros Christodoulakis Technical University of Crete Laboratory of Distributed Multimedia Information Systems and Applications (MUSIC) Kounoupidiana, 73100 Chania, Greece allegra@ced.tuc.gr, stavros@ced.tuc.gr

Abstract

The OntoNL Framework provides an architecture and re-usable components for automating as much as possible the building of natural language interfaces to information systems. In addition to the syntactic analysis components, OntoNL has semantic analysis components which exploit domain ontologies to provide better disambiguation of the user input. We present in this paper the algorithms used for semantic processing of the natural language queries, as well as an ontology-driven semantic relatedness measure developed for this purpose. We also present extensive evaluation results with different ontologies using human subjects.

1. Introduction

Natural language interfaces to information systems have significant advantages. It is however well known that a major problem in their use is the existence of ambiguities in the user interactions which lead to lengthy disambiguation dialogues.

In this paper we present the semantic processing of natural language queries in the OntoNL software engineering Framework, in an attempt to tackle the problem. The OntoNL is an ontology-based natural language interface generator to knowledge repositories. We aim to reduce the ambiguities in the user interactions by providing domain specific semantic analysis. In comparison with natural language interfaces that focus either on developing methodologies only for syntactic analysis or for a specific application, the OntoNL Framework handles syntactic and semantic ambiguity at both a general and a domain specific environment and represents in an ontology query language, disambiguated and ranked natural language expressions. The purpose of the semantic disambiguation, based on a particular domain is to eliminate the possible senses that can be assigned to a word in the discourse and associate a sense which is distinguishable from other meanings.

In the OntoNL that focuses on disambiguation utilizing semantic processing, we could not rely on a general hierarchy of terms like the WordNet [2] or the general ontology SUMO to disambiguate user expressions. We propose a method that can be used for computing semantic relatedness between concepts that constitute domains of context and are described by OWL domain ontologies.

The semantic ranking procedure proposed is designed to clarify sense ambiguities. The procedure uses information from the ontologies and the specific clusters of context inside an ontology. Given an OWL ontology, weights are assigned to links between concepts based on certain properties of the ontology, so that they measure the level of relatedness between concepts. In this way we can identify related concepts in the ontology that guide the semantic search procedure. An important property of the relatedness measure is that it is asymmetric (the relatedness between A and B does not imply the opposite) since relations that are described with natural language do not indicate mathematical rules. The semantic relatedness is used for the determination of the optimum, most related path that leads from the source concept-subject part to the target concept-object part of a natural language expression.

The result of the syntactic and semantic processing in the OntoNL Framework is SPARQL queries that reflect the natural language query and are ranked using the semantic relatedness measure.

The Framework has been implemented and we have performed extensive experimentation with human subjects to fine tune the parameters of the system and to evaluate the performance of the semantic ranking in different domains with different domain ontologies.

2. The OntoNL Framework

The OntoNL Software Engineering Framework has two major objectives. The first is to minimize the cost of build-

ing natural language interfaces to information systems by providing reusable software components that can be used in different application domains and knowledge bases, and adapted with a small cost to a new environment. The second is to do semantic processing, exploiting domain ontologies in order to reduce ambiguities in a particular domain.



Figure 1. The OntoNL Framework Architecture

The architecture of the Framework is shown in figure 1. The Framework in a particular application environment has to be supplied with domain ontologies (encoded in OWL) which are used for semantic processing. The user input in an application environment may be

- Requests for metadata (ex. Show me the goals scored in the game between Italy and France)
- WH-questions (ex. What was the score in the game between Italy and France?)
- Yes/No questions (ex. Were there any goals in the game between Italy and France?)

The output for a particular input NL query is a set of one or more weighted disambiguated to the specific domain queries, encoded in SPARQL. We choose SPARQL as the query language to represent the natural language queries since SPARQL is defined in terms of the W3C's RDF data model and will work for any data source that can be mapped into RDF. If the environment uses a different type of repository than OWL-SPARQL, a module has to be implemented that does the mapping from the SPARQL encoded queries to the schema and query language that the environment uses (Relational Schema-SQL, XML Schema-XQUERY, etc). Since this transformation is Schema dependent it is not automated within the Framework software.

The main components of the OntoNL provide Linguistic Analysis and Ontology Processing for Semantic Disambiguation. The Linguistic Analysis includes components for user input conversion (we cut-off the unnecessary information for the retrieval and provide a form that meets the OntoNL Expressions model [6]), Part-Of-Speech tagging, Noun Compound Bracketing, Grammatical Relations Discovery, and Synonym and Sense Discovery by using input from the WordNet which provides information about word synonyms. More details of the methodologies used for linguistic analysis can be found in [8].

The Semantic Disambiguation Module of the OntoNL is responsible for domain specific disambiguation and result ranking. It is described in more detail in the next section.

The Linguistic Analysis module produces instances of the OntoNL Expressions model. These instances are used by the Ontology Processor for semantic disambiguation and ranking of resulted SPARQL queries.

3. The OntoNL Semantic Disambiguation Algorithm

As we already stated the semantic disambiguation is a process that targets to eliminate the possible senses that can be assigned to a word in the discourse, and associate a sense which is distinguishable from other meanings. However, WordNet gives only generic categories of senses and not domain specific. Thus it is clear that much better semantic disambiguation can be done when domain knowledge is available in the form of ontologies. The purpose of the OntoNL Semantic Disambiguation Module is to use information of the OntoNL Ontology Processor in the OntoNL Framework (figure 1) in order to do semantic disambiguation of the natural language queries. The input in the Ontology Processor are instances of the expressions model produced by the Syntactic Analyzer, which include terms extracted from the natural language input, their synonyms, and their tagging according to the expressions model constructs and OWL Ontologies. The output is disambiguated sentences expressed as queries in SPARQL, or in the case that complete disambiguation is not possible, a set of ranked SPARQL queries.

In particular, the common types of ambiguity encountered in the OntoNL Framework are:

- The natural language expression contains general keywords that can be resolved by using only the ontology information (ontological structures and semantics). For example, in the expression "*players of soccer team Milan*", the words *players* and *soccer team* are matched to the corresponding concepts of the domain ontology and the information that *Milan* is the name of a soccer team comes from the syntax of the natural language expression (object complement that follows a direct object)
- · One of the subject/object part of the expressions model

contains terms that cannot be disambiguated by using the ontology. For example, in the expression " *the players of Barcelona*", the word *players* is matched to the corresponding concept of the domain ontology but the system cannot "understand" the sense of the object part of the natural language expression, *Barcelona* that could be a soccer team, a city etc. The algorithm "considers" the word *Barcelona* as a concept instance, an individual.

• Neither the subject nor the object part contains terms disambiguated by using the ontological structures. For example, in the expression "*information about Milan*", neither the word *information* nor the word *Milan* are matched to the ontological structures of the domain ontology. The system "considers" the word *information* as an unresolved concept and the word *Milan* as an unresolved concept instance.

Figure 2 shows the general steps of the semantic disambiguation algorithm used in OntoNL using UML Activity Diagram notation. The approach is general for any OWL DL or Full domain ontology.



Figure 2. The OntoNL Semantic Disambiguation procedure

The algorithm searches to see if there is a correspondence between the naming of the expressions model instance and the ontological structures. If there is a complete match, a Relatedness Value measure is assigned with value 1 to indicate the complete relevance of the sentence with the specific domain. If the disambiguation is not complete for the Object Part, the algorithm checks for the number of the terms that show ambiguity. If the ambiguity is in the Subject Part then the algorithm checks for a number specified by the application of ontology concepts that have the greatest relatedness value with the disambiguated term of the request. If there is only one term with an ambiguity or the ambiguity is in the Subject Part, the algorithm checks and retrieves

the output of the OntoNL Ontologies Processor for a number, specified by the application, of the most related concepts to the concept that comprise the subject or the object part (if the ambiguity is in the object or the subject part respectively) of the expression. If in the object part are more than one terms with ambiguities, the algorithm checks for operators (or/and). In the existence of an operator the algorithm considers the terms to be concept instances of the same concept of the domain ontology. In the absence of an operator the algorithm considers the terms to be concept instances of a different ontology concept. Then the algorithm searches for a number, specified by the application, of the most related concepts to the concept that found a correspondence to the ontological structures and assigns the relatedness measure, already calculated by the OntoNL Ontologies Processor. The last activity of the algorithm is to enhance the Ontology Structure class of the OntoNL Expressions Model with the corresponding ontology concepts to natural language terms in the class attribute and with the relatedness measurement value the value attribute.

4. The OntoNL Semantic Relatedness Measure

When a query cannot be disambiguated completely from the OntoNL Semantic Disambiguation procedure, OntoNL uses a Semantic Relatedness Measure [6] to suggest weighted possible interpretations of the user request. To that purpose, OntoNL borrows and expands ideas from the research of Semantic Relatedness of concepts in semantic networks.

The Relatedness Matrix contains a weight of relatedness (Relatedness Measure) between any two concepts. The relatedness measure depends on the semantic relations defined by properties in OWL. Properties can be used to state relationships between individuals (named Object-Properties) or from individuals to data values (named DatatypeProperties).

The algorithm also takes into account the semantic relation of OWL: EquivalentClass. The class that is OWL: EquivalentClass with a source class has a similarity (not relatedness) value 1. In our computations, the classes related to the source class of the ontology are also related with the same value to the equivalent class.

In all other cases the relatedness value computation is based on the following factors: the commonality (based on the semantic relations and the conceptual distance) and the related senses.

The *commonality* depends on the amount of the common information two concepts share. The commonality measure has two factors: The position of the concepts relatively to the position of their most specific common subsumer/fatherclass(will be examined by the conceptual distance and the specificity measurement) and the reciprocity of their properties (if the connecting OWL ObjectProperties have also inverse properties).

Based on the semantic relations when we detect that a source concept-class is immediately related via an Object-Property with the target concept, the relatedness value is set to 1. Else, we count the number of the common properties of the two concepts and the number of the common properties the two concepts share that are inverseOf properties:

$$rel_{prop}(c_1, c_2) = (f_1 \times \frac{\sum_{i=1}^n p_{i12}}{\sum_{i=1}^n p_{i1}}) + (f_2 \times \frac{\sum_{i=1}^n p_{invi12}}{\sum_{i=1}^n p_{i12}}),$$
(1)

where

$$f_1 \ge f_2, f_1 + f_2 = 1, (f_1, f_2) = weights$$
 (2)

In the above equations, the value p_{i1} represents the fact that concept c_1 is related to concept c_i (value: 0 or 1 in general). The value p_{i12} represents the fact that both concepts c_1 and c_2 are related to concept c_i . The p_{invi12} represents the fact that both concepts are inversely related.

The *conceptual distance* measure is based on two factors; the path distance and the specificity. The path distance measures the relatedness of two concepts by counting the minimal path of edges between the two concepts through their structural relations (IS-A relations). The OntoNL disambiguation algorithm uses the relatedness of concepts of the domain ontologies and not the similarity, so the measure excludes the cases were $d_{C1} = 0$, $d_{C2} = 0$ and $d_{C1} + d_{C2} = 2$. So, the value of distance is calculated as:

$$\forall d_{C1} \ge 1, d_{C2} \ge 1, d_{C1} + d_{C2} > 2:$$

$$pathDist(c_1, c_2) = \frac{d_{C1} + d_{C2}}{2 * D} \in (0, 1]$$
(3)

where d_{C1} is the number of edges from concept 1 to the closer common subsumer and d_{C2} the number of edges from concept 2 to the closer common subsumer. D is the maximum depth of the ontology.

The parameter that differentiates our measure from the classic measures of distance counting is that it is combined with the *specificity* factor. When the change of direction (from superClassing to subClassing and opposite) is close to the concept/subject of the OntoNL Expressions model($d_{C1} \ll (d_{C1}+d_{C2})/2$), the two concepts are more related. When the direction of the path changes far from the reference concept then the semantics change as well (more specialization of the reference concept).

We count the specificity of the concepts inside the ontology by the following normalized weight value:

$$w1_{specC1} = -\ln \frac{2 \times d_{C1}}{d_{C1} + d_{C2}} \in (0, 1], \text{ if } d_{C1} < \frac{d_{C1} + d_{C2}}{2}$$

else

$$w1_{specC1} = 0, ifd_{C1} \ge \frac{d_{C1} + d_{C2}}{2}$$
 (4)

We, also propose a method of counting the specialization of the concept - c1 based on the object properties of the subsumer, by the factor:

$$spec_{C1} = \frac{\#ObjP_{C1} - \#ObjP_S}{\#ObjP_S} \in [0,\infty)$$
 (5)

where Obj_{PC1} is the number of Object Properties of the concept C_1 and Obj_{PS} is the number of ObjectProperties of the subsumer concept. The range of the $spec_{C1}$ is $[0, \infty)$. To limit the range in [0, 1] we need to restrict the number of ObjectProperties of the concept C_1 :

$$for #ObjP_{C1} \le 10 \times #ObjP_S:$$

$$w2_{specC_1} = 1 - \log \frac{#ObjP_{C1}}{#ObjP_S} \in [0, 1]$$
(6)

The conceptual distance measure then becomes:

$$rel_{CD} = (w1_{specC_1} + w2_{specC_1} + 1 - pathDist(c_1, c_2))/3$$
(7)

The *related senses* measure counts the common senses of two concepts by counting the common nouns and synonyms extracted from the descriptions of the concepts in the ontology (owl:label, owl:comment) or from the descriptive part of the term meaning in the WordNet. Let S_1 be the description set of senses for c_1 and S_2 the description set of senses for c_2 . The related senses measure is:

$$rel_{RS}(c_1, c_2) = \frac{|S_1 \cap S_2|}{|S_1 \cap S_2| + |S_1 \setminus S_2|}$$
(8)

where S_1 is the description set of senses for c_1 and S_2 the description set of senses for c_2 .

The overall relatedness measure is the following:

$$forw_{1} + w_{2} + w_{3} = 1, (w_{1}, w_{2}, w_{3}) > 0,$$

$$rel_{PROP}(c_{1}, c_{2}), rel_{CD}(c_{1}, c_{2}), rel_{RS}(c_{1}, c_{2}) \in [0, 1]:$$
(9)

$$rel_{OntoNL} = w_{1} \times rel_{PROP} + w_{2} \times rel_{CD} + w_{3} \times rel_{RS}$$
(10)

The three factors w_1 , w_2 and w_3 , help of balancing among the parameters depending on the application ontology. The measure is applied in all concepts of the ontology in the preprocessing phase and constructs the Relatedness Matrix, a NxN matrix (N is the total number of concepts) with the relatedness values of each concept against all the other concepts of the disambiguation ontology.

5. Experimental Evaluation

A complete evaluation framework has been designed for the OntoNL generator. As far as it concerns the OntoNL Semantic Relatedness Measure evaluation, the framework takes into account a large number of parameters regarding the characteristics of the ontologies involved and the types of users.

We have focused our attention to the performance experimentation in a generic way utilizing readily available ontologies in the web, not carefully constructed by hand ontologies. As we discussed in the previous section the three factors w_1, w_2 and w_3 of the overall OntoNL measure help of balancing among the three sub-measure depending on the application ontology. We need to bound their values and provide the complete measurement that will show good results regardless of the OWL ontology used. In order to assess the impact of each of the sub-measures we needed to evaluate it against a "gold standard" of object relatedness. To that end we designed a detailed experiment in which human subjects were asked to assess the relatedness between pairs of objects (preliminary results can be found in [6]) and afterwards the precision of the OntoNL Semantic Relatedness measure in a specific application, for the domain of soccer, a context familiar with the users.

5.1. Study Design

Experiments that rely on human judgments have become the benchmark in determining the similarity of words in NLP research [1, 4, 10]. We reused the overall experimental design of these studies and adapted it to be usable for complex objects in an ontology. We proceeded as follows: First, we have found a number of suitable object pairs from a number of ontologies freely available in the web fulfilling the following criteria:

- At least two pairs from each ontology should be in close vicinity in the ontology-graph.
- At least two pairs from each ontology should be far apart in the ontology-graph.
- At least one pair from each ontology should consist of a concept and its descendant/specialization.

The rest of the concepts were paired in a way such that the concepts' name, description, attributes, or properties(e.g., parts) featured some relatedness. We point out that the subjects' ability to relate to the ontology content (domain) was crucial for the success of the experiment.

We have obtained relatedness judgments from 25 human subjects, 10 from the computer science field that had some knowledge of the domain ontologies and 15 from the liberal arts field, that were used for the evaluation, for 85 pairs of concepts that we meet in seven OWL domain ontologies¹

freely available on the web, for different domains. For each ontology we had 10 data points except the Soccer Ontology that we had 25 data points because of its size and richness in properties. The main goal was to compare the OntoNL submeasures and the overall measure on how well they reflect human judgments. The subjects had the opportunity during the evaluation, to see the properties and the description (if any) of the concepts that they had to assess the relatedness.

The pairs ranged from "highly related" to "semantically unrelated", and the subjects were asked to rate them, on the scale of 0.0 to 1.0, according to their "relatedness of meaning". The users were asked to specify how they had made the assessment: 1. by concept name, 2. by concept description, 3. by concept properties, 4. a combination of 1-3, and 5. using other assessment methods. This question had as objective to find which features were used by the subjects in their evaluation- a notion that similarity researchers in the social sciences have found to be central [3], but also useful to us for the determination of the impact of the submeasures of relatedness to the overall measure. Then, we asked the subjects to apply requests for a specific domain, the domain of soccer. We gathered a total of 40 requests concerning the second type of ambiguity of the disambiguation algorithm, after eliminating any duplicates.

5.2. Results

Our first objective was to investigate what are the values of the parameters f_1 , f_2 , w_1 , w_2 , w_3 for each ontology, and overall. We did not use different data sets for the parameter setting and the main evaluation. Human subjects were used for the experiments. We observed that the optimal values of these parameters strongly depend on the ontology. Their optimal experimental values are shown in Table 1.

Table 1. The values of the relative weights f_1 and f_2 of eq. 3 and w_1 (for rel_{PROP}), w_2 (for rel_{CD}) and w_3 (for rel_{RS}) of eq. 11 for the test set of ontologies

Ontology	rel _{PROP}		rel _{OntoNL}		
	f_1	f_2	w_1	w_2	w_3
Soccer Ontology	0,5	0,5	0,7	0,2	0,1
Wine Ontology	0,65	0,35	0,5	0,25	0,25
People Ontology	0,1	0,9	0,45	0,2	0,35
Pizza Ontology	0,65	0,35	0,5	0,27	0,23
Koala Ontology	0,99	0,01	0,25	0,65	0,1
Images Ontology	0,33	0,67	0,45	0,5	0,05
Travel Ontology	0,9	0,1	0,7	0,1	0,2

We observe that w_1 and f_1 are in general the most important of the weights, which implies that the number of common

Travel:http://learn.tsinghua.edu.cn/home-page/2003214945/travelontology.owl Wine:http://www.w3.org/TR/owl-guide/wine.rdf

¹Images:http://www.mindswap.org/glapizco/technical.owl Koala:http://protege.stanford.edu/plugins/owl/owl-library/koala.owl People:http://owl.man.ac.uk/2005/07/sssw/people.html Pizza:http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl

Soccer:http://lamia.ced.tuc.gr/ontologies/AVMDS03/soccer

properties of two concepts is a significant factor in determining the relatedness. The conceptual distance measure (w_2) and the related senses measure (w_3) seem to have also significant impact, but in almost all ontologies (except the Koala and Images ontologies) the impact of each one of them was less than the common properties measure. Among these two measures the related senses measure (w_3) had a stronger impact than the conceptual distance measure (w_2) in two ontologies, while the conceptual distance measure (w_2) had a stronger impact in four ontologies.

Using the optimal values, computed manually, for the parameters we studied how the computed relatedness measure among two concepts was correlated (using Pearson's correlation) with the relatedness perceived by the human subjects. Table 2 shows the computed correlation coefficients with relative weights of Table 1 between the system computed relatedness measure and the human subjects evaluated relatedness (0,7 * relValue from the Liberal Arts Field subjects).

Table 2. The values of the coefficients of correlation between human ratings of relatedness and the OntoNL Semantic Relatedness sub-measures and overall measure

Human Subjects Ratings				
Measure	rel _{PROP}	rel_{CD}	rel_{RS}	rel _{OntoNI}
Soccer Ontology	0,910	0,594	0,329	0,943
Wine Ontology	0,832	0,644	0,830	0,976
People Ontology	0,906	0,937	0,949	0,984
Pizza Ontology	0,657	0,77	-	0,863
Koala Ontology	0,492	0,846	0,285	0,857
Images Ontology	0,964	0,953	0,273	0,997
Travel Ontology	0,946	0,891	0,612	0,973

The results are satisfactory and show that the average OntoNL measure correlation for each ontology was almost always more than 0.9 and in 4 out of the 7 cases they were more than 0.95. The average correlation was 0.94. In all cases the calculated by the system weighted relatedness measure was higher correlated with the human subject evaluations than the correlations of the partial semantic measures (common properties, related senses, conceptual distance).

Table 3. The values of the relative weights f_1 and f_2 of eq. 1 and w_1 (for rel_{PROP}), w_2 (for rel_{RS}) and w_3 (for rel_{CD}) of eq. 13

	rel _{PROP}		rel _{OntoNL}		
OWL Domain Ontologies	f_1	f_2	w_1	w_2	w_3
-	0,65	0,35	0,5	0,27	0,23

An observation mentioned above was the relatively large

variability of the optimal weights for each ontology. We decided to experiment with the same set of weights for all the ontologies (the optimal values for the Pizza Ontology), to observe if the relatedness measures were drastically affected, and if they are still satisfactory. Table 3 shows the common set of weights used for all the experiments with all the ontologies.

Human Subjects Ratings				
Measure	rel _{OntoNL}	rel _{OntoNL} '		
Soccer Ontology	0,943	0,918		
Wine Ontology	0,976	0,974		
People Ontology	0,984	0,966		
Pizza Ontology	0,863	0,863		
Koala Ontology	0,857	0,798		
Images Ontology	0,997	0,973		
Travel Ontology	0,973	0,935		

Table 4. A comparison of the correlations between human ratings of relatedness and the overall OntoNL measure with relative weights of Table 1 and Table 3

Table 4 shows the correlations obtained between the system computed values and the human subject computed values (second column). For comparison reasons the first column shows the correlations computed with different weights of Table 1 (copied from Table 2). Table 4 shows that the results obtained, as expected, are worse than the results obtained using different weights for each ontology. The correlation however between human subject and the system evaluations, are quite high. The average drop in correlation was 0.024, while the maximum drop in one ontology was 0.06. In this case (Koala Ontology) the average correlation dropped below 0.8 (to 0.798). For this ontology however, even with its optimal weights the correlation was not very high (0.863).



Figure 3. The effectiveness of ontology mappings to user input

For the application-based evaluation we have presented to the subjects the resulted concepts related to the subject concept of their request ranked based on the semantic similarity measurement. We are interested in the success ranking of the related concepts in each particular domain and this is what we measured. The users replied the ranking position of their correct response in mind and this experiment was conducted twice. The figure 3 shows the number of requests in the top percentage of the total number of results. In the first iteration we get almost 90% of the exact result in the 60% of the total number of results. In a second iteration we get more than 90% of the exact result in the 50% of the total number of results. The conclusion that derives is that in a second iteration of tests the results were more accurate because the familiarity of using the system increased.

5.3. Discussion

Our research is ongoing and focuses in developing an automatic method for determining the weights for any given ontology. We are developing metrics that we have found that affect the choice of weight like:

Object Property Richness: This metric reflects the placement of properties in an OWL ontology. An ontology that contains many object properties is richer than a taxonomy with only class-subclass relationships. The number of object properties that are defined for each class can indicate both the quality of ontology design and the amount of information pertaining to instance data.

Specificity Richness: This metric describes the specialization of information across different levels of the ontology's inheritance tree. This is a good indication of how more descriptive knowledge becomes by moving vertical in the ontology.

Inheritance Richness: This metric describes the distribution of information across different levels of the ontology's inheritance tree. This is a good indication of how well knowledge is grouped into different categories and subcategories in the ontology. This measure can help on distinguish a horizontal from a vertical ontology.

Readability: This metric indicates the existence of human readable descriptions in the ontology, such as comments, labels, and captions. This metric can be a good indication if the ontology is going to be queried and the results are going to be listed to users.

The category of metrics we are interested in is the schema metrics that evaluates ontology design and its potential for rich knowledge representation. The metrics we are proposing are not 'gold standard' measures of ontologies. Rather than describing an ontology as merely effective or ineffective, metrics describe a certain aspect of the ontology because, in most cases, the way the ontology is built is largely dependent on the domain in which it is designed. The conclusion here is that the readability and the richness of the domain ontology is a key aspect for the successful use of the OntoNL Semantic Relatedness Measure.

6. Related Work

The natural language interaction research community has recently started to explore semantic disambiguation using ontologies, in systems like AQUA [12]. AQUA translates English questions into logical queries and is coupled with the AKT reference ontology for the academic domain, written in OCML. The system works in a domainspecific pattern-matching mode trying to find exact matches with names in the specific ontology and has not tested its reusability in different domains.

An attempt of using a more knowledge oriented approach for the construction of a natural language interface for the domain of digital TV is described in [7]. It used both ontologies and User Profiles in order to do semantic natural language processing. Ontologies used for this system were capturing the TV-Anytime standard. The approach was tuned and depended on the specific ontologies and lacked the generality and the completeness of the system described in this paper. In addition the domain ontologies were based on keywords and not on deep knowledge structures. All the above systems lack the robustness and the reusability of a software engineering framework.

The NLP literature provides the largest group of related work for measuring semantic relatedness that in most cases are based on lexical resources or WordNet and other semantic networks or deal with computing taxonomic path length.

A simple way to compute semantic relatedness in taxonomies like WordNet is to view it as a graph and identify relatedness with path length between the concepts [11]. This approach was followed in other networks also, like the MeSH (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=mesh), a semantic hierarchy of terms used for indexing articles in the bibliographic retrieval system MEDLINE, by Rada et al., [10].

Jiang and Conrath [5] propose a combined edge counting and node based method that outperforms either of the pure approaches. This hints at the usefulness of combined approaches like the OntoNL Semantic Relatedness Measure we propose in this paper. The research that was made by Budanitsky and Hirst [1] support our claim that the quality of similarity measures is dependent on the ontology in general. They find that differences in the quality of WordNetbased similarity measurement algorithms found in various papers can be explained by the different versions of Word-Net that have been used.

To confront with this issue Lin [9] tries to develop an information-theoretic measure of similarity that is not tied to a particular domain or application and that is less heuristic in nature with success. The measure is slightly found to excel Resnik's similarity algorithm [11]. The drawback is that it still requires a probabilistic model of the applica-

tion domain, retrieved by parsing a large word corpus. This limitation makes it problematic to smaller ontologies.

All the research results presented in the literature so far [1, 4, 5, 10, 11] were tested on specific ontologies like the WordNet and MeSH (http://www.nlm.nih.gov/mesh/) ontologies, they are not general and have not been tested in different domain ontologies that refer to different contexts. The WordNet and MeSH ontologies are well formed hierarchies of terms and the methodologies that have used them examined basically similarity between terms, like the similarity between *car* and *automobile*, *autograph* and *signature* [1], and not relatedness between concepts, like the relatedness in a particular domain of *goals* and *football player*, *team* and *coach*. Also, most of these approaches are focused on the comparison of nouns, limiting their generality to complex objects or even hierarchies of verbs.

7. Conclusions

We have presented the OntoNL Ontology-driven Semantic Relatedness measure for OWL ontologies used for natural language disambiguation in the OntoNL Framework. The OntoNL software engineering Framework is used for the generation of natural language user interfaces to knowledge repositories. The methodology uses domain specific ontologies for the semantic disambiguation. The ontologies are processed offline to identify the strength of the relatedness between the concepts. Strongly related concepts lead to higher ranked pairs of concepts during disambiguation. The measure is based on the commonality of two concepts, the related senses that may share, their conceptual distance in the ontology and their specificity in comparison with their common root concept. The number and the semantics of the properties that specialize a concept of an OWL ontology over other concepts helped the construction and the effectiveness of the OntoNL Semantic Relatedness Measure. The conceptual distance is also a measure that has a great influence if the ontology depth is big because of the several paths that lead from the source concept (that is the subject part of a natural language expression) to the target concept (that is the object part of a natural language expression.

The motivation of this work came from the absence of a general, domain-independent semantic relatedness measure apart from the WordNet. The measure was successfully used for natural language disambiguation and semantic ranking in the OntoNL Framework. The disambiguation process depends on the domain ontologies and when necessary, the OntoNL Semantic Relatedness Measure is used to rank ontological, grammatically-related concepts.

We have developed an evaluation framework for the OntoNL Natural Language Interface Generator. For the OntoNL Semantic Relatedness Measure evaluation, the framework takes into account a number of parameters regarding the characteristics of the ontologies involved and the types of users. We have focused our attention to the performance experimentation in a generic way utilizing readily available ontologies in the web, not carefully constructed by hand ontologies. The very good and promising results of the experiments with 7 OWL domain ontologies, freely available on the web are presented as a comparison of the measurement of relatedness between human subjects and the OntoNL measure. Also, a specific application of the measure showed good precision and user satisfaction.

For future improvements, we may need to investigate the influence of more complex structures of OWL vocabulary to the performance.

References

- A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.
- [2] C. Fellbaum. WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, MA, 1998.
- [3] D. Gentner and J. Medina. Similarity and the development of rules. *Cognition*, 65:263–297, 1998.
- [4] M. Jarmasz and S. Szpakowicz. Roget's thesaurus and semantic similarity. In *RANLP04 Conference Proceedings*, pages 212–129, 2003.
- [5] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, 1997.
- [6] A. Karanastasi and S. Christodoulakis. Ontology-driven semantic ranking for natural language disambiguation in the ontonl framework. In *Proceedings of the 4th European Semantic Web Conference (ESWC)*, 2007.
- [7] A. Karanastasi, F. Kazasis, and C. Christodoulakis. A natural language model and a system for managing tv-anytime information in mobile environments. ACM Verlag International Journal of Personal and Ubiquitous Computing, 9(5):262– 272, 2005.
- [8] A. Karanastasi, A. Zwtos, and S. Christodoulakis. User interactions with multimedia repositories using natural language interfaces - ontonl: an architectural framework and its implementation. *Journal of Digital Information Management -JDIM*, 4(4):220–226, 2006.
- [9] D. Lin. An information-theoretic definition of similarity. In Proceedings of the 15th International Conference on Machine Learning, 1998.
- [10] R. Rada and E. Bicknell. Ranking documents with a thesaurus. JASIS, 40(5):304–310, 1998.
- [11] P. Resnik. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.
- [12] M. Vargas-Vera and E. Motta. Aqua ontology-based question answering system. In *Proceedings of the Third International Mexican Conference on Artificial Intelligence*, 2004.