The OntoNL Framework for Natural Language Interface Generation and a Domain-specific Application

Anastasia Karanastasi, Alexandros Zotos, Stavros Christodoulakis
Laboratory of Distributed Multimedia Information Systems / Technical University of Crete (MUSIC/TUC)
University Campus, Kounoupidiana, Chania, Greece
{allegra, azotos, stavros}@ced.tuc.gr

Abstract

We present in this paper the design and implementation of the OntoNL Framework, a natural language interface generator for knowledge repositories, as well as a natural language system for interactions with multimedia repositories which was built using the OntoNL Framework. The system allows the users to specify natural language requests about the multimedia content with rich semantics that result to digital content delivery. We propose and evaluate a semantic relatedness measure for OWL domain ontologies that concludes to the semantic ranking of ontological, grammatically-related structures. This procedure is used to disambiguate in a particular domain of context and represent in an ontology query language, natural language expressions. The ontology query language that we use is the SPARQL. The construction of the queries is automated and also dependent on the semantic relatedness measurement of ontology concepts. The methodology has been successfully integrated into the OntoNL Framework. The experimentations show a good performance in a number of OWL ontologies.

Categories and Subject Descriptors

H.1.2 [User/Machine Systems]: Human factors, I.2.1 [Applications and Expert Systems]: Natural language interfaces, I.2.4 [Knowledge Representation Formalisms and Methods]: Semantic networks, I.2.7 [Natural Language Processing]: Language parsing and understanding.

General Terms

Algorithms, Measurement, Design, Experimentation, Human Factors, Languages

Keywords

Natural language interfaces; ontologies; semantic relatedness; query representation

1 Introduction

The need to determine semantic relatedness between two lexically expressed concepts is a problem that concerns natural language processing for a long time now. Measures of relatedness or distance are used in applications of natural language processing as word sense disambiguation, determining the structure of texts, information extraction and retrieval and automatic indexing.

It is also well known that a problem with the natural language interfaces to information repositories is the ambiguities of the requests, which may lead to lengthy clarification dialogues. Due to the complexity of natural language, reliable natural language understanding is an unaccomplished goal in spite of years of work in fields like Artificial Intelligence, Computational Linguistics and other. The natural language understanding could be approached by applying methods for consulting knowledge sources such as domain ontologies. Ontologies are usually expressed in a formal knowledge representation language so that detailed, accurate, consistent, sound, and meaningful distinctions can be made among the classes (general concepts), properties (those concepts may have), and the relations that exist among these concepts. A module dealing with ontologies can perform automated reasoning using the ontologies, and thus provide advanced services to intelligent applications such as: conceptual/semantic search and retrieval, software agents, decision support, speech and natural language understanding and knowledge management.

Knowing the context in which an ambiguity occurs is crucial for resolving it. This observation leads us to try to exploit domain ontologies that describe the domain of use of the natural language interface. The methodology that we have developed is reusable, domain independent and works with input only the OWL ontology that was used as a reference schema for constructing a knowledge repository. The methodology depends on a semantic relatedness measure that we have developed for domain ontologies that concludes to semantic ranking. The semantic ranking is a methodology for ranking related concepts based on their commonality, related senses, conceptual distance, specificity and semantic relations. This procedure concludes to the natural language representation for information retrieval using an ontology query language, the SPARQL. The SPARQL queries are ranked based on the semantic relatedness measure value that is also used for the automatic construction of the queries.

This methodology is integrated in the OntoNL Framework, a natural language interface generator to knowledge repositories. We present the OntoNL Framework for building natural language interfaces to semantic repositories, as well as the NL2DL, a natural language interaction interface for semantic multimedia repositories which was built using the OntoNL Framework. The application of the OntoNL Framework addresses a semantic multimedia repository with digital audiovisual content of soccer events and metadata concerning soccer in general, has been developed and demonstrated in the 2nd and 3rd Annual Review of the DELOS II EU Network of Excellence (IST 507618) (http://www.delos.info/).

The OntoNL Framework implements a software platform that automates to a large degree the construction of natural language interfaces for knowledge repositories. To achieve the applicability and reusability of the OntoNL Framework in many different applications and domains, the supporting software is independent on the domain ontologies.

The software components of the OntoNL Framework address uniformly a range of problems in sentence analysis each of which traditionally had required a separate mechanism. A single architecture handles both syntactic and semantic analysis, handles ambiguities at both the general and the domain specific environment. At the same time, the Framework has been designed in a way to avoid dependencies with the information repository so that it becomes reusable in different applications with different domain semantics.

2 The OntoNL Framework

The OntoNL Software Engineering Framework has two major objectives. The first is to minimize the cost of building 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. The output of a natural language request is a ranked set of queries in an ontology query language.

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 is natural language requests and WH-sentences (who, were, what, etc.). 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 POS tagging, Noun Compound Bracketing, Grammatical Relations Discovery, and Synonym and Sense Discovery. To perform its functions it uses input from the WordNet which provides information about word synonyms and the part of speech that a word is (verb, noun, etc.). 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.

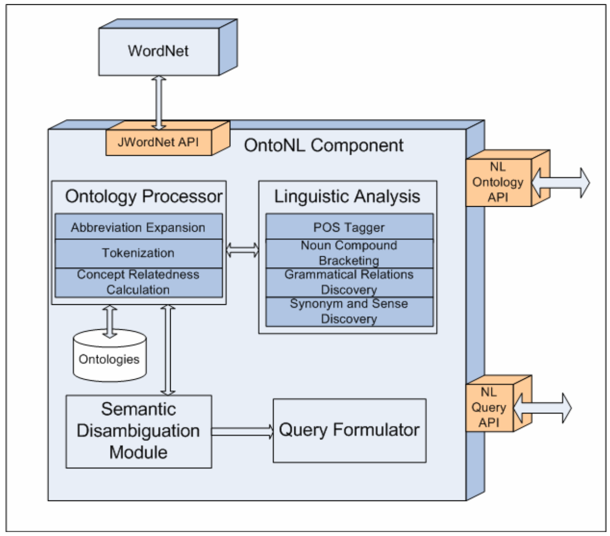

Figure 1. The OntoNL Framework Architecture

The language model used in OntoNL supports both the Linguistic Analyzer and the Ontology Processor. The language model is shown in figure 2 using UML Class Diagram notation. There are lists of words that constitute the basic sentence structures, like the subject and the object and there are complements and special cases of objects that predicate them.

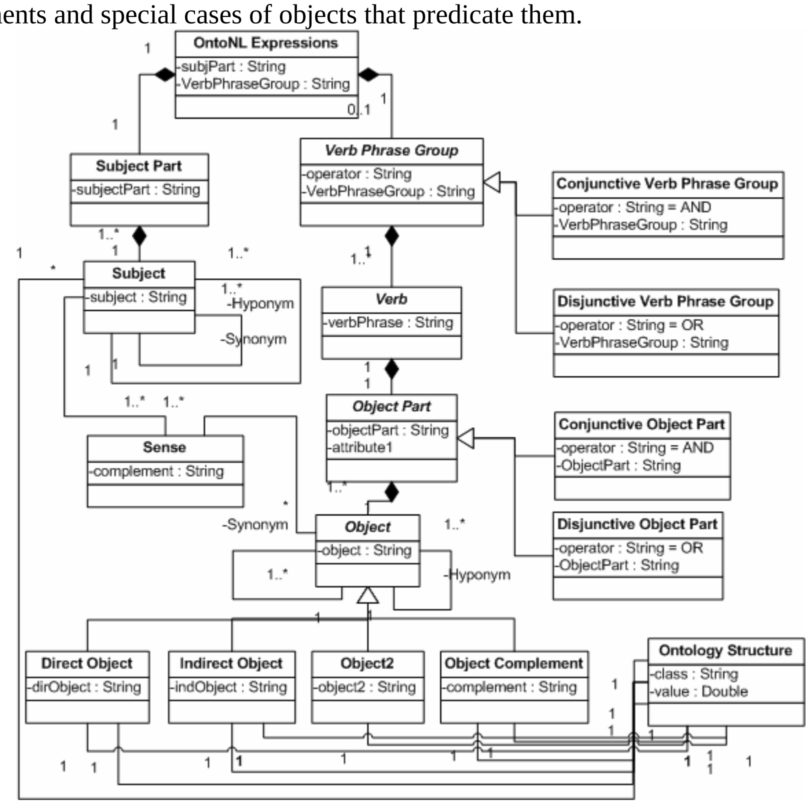

Figure 2. The OntoNL Language Model that derives from the syntactic and semantic analysis, based on the OntoNL Natural Language Expressions

The OntoNL Expressions is the general class that summarizes the cases of possible grammatical dependencies inside an utterance. It consists of a Subject Part and possibly of a Verb Phrase Group. The Subject Part consists of one or more Subjects that are connected with a Boolean Or operator. The Verb Phrase Group is an abstract class and has an IS-A relation with the Conjunctive and the Disjunctive Verb Phrase Group. This is done in order to distinguish the cases where the Subject

Part can be described by more than one verb phrases that are separated by Boolean Operators. The Verb Phrase Group consists of one or more Verbs. The Verb consists of a unique Object Part, an abstract class of one or more Objects. Again the Object Part has an IS-A relation with the Conjunctive and the Disjunctive Object Part. That means that after the verb there may be more than one object parts jointed with a Boolean AND or OR operator. The Object can be a Direct Object, an Indirect Object, an Object Complement or a combination of these.

The Syntactic Analyzer produces instances of this language model. These instances are used by the ontology processor for semantic disambiguation and ranking of results.

2.1 The OntoNL Linguistic Analyzer

The analysis consists of the following parts:

Part-Of-Speech (POS) Tagging. In our system we use the Stanford Log-Linear POS Tagger (nlp.stanford.edu/software/tagger.shtml) as the first step of the disambiguation process.

Noun Compound Analysis. The OntoNL expands noun compound n-grams into all possible morphological forms. It uses the dependency model [Layer 1995] for syntactically analyzing noun compounds. The corpus is provided by the domain ontologies and consists of the concept names that are compound nouns, their synonyms from the WordNet, and the <owl:label> and the <owl:comment> content of the OWL ontology.

Sentence Patterns: Locating Grammatical Relations. Grammatical relation detection is the semantic basis for the information extraction.

We developed an annotation scheme for locating grammatical relations. The scheme is based on grammatical relations that are composed of bilexical dependencies (between a head and a dependent) labeled with the name of the relation involving the two words. The grammatical relations are arranged in a hierarchy (see figure 3), rooted with the most generic relation, dep(dependent).

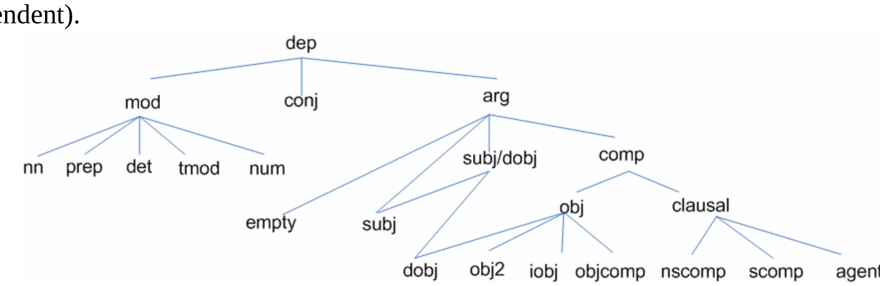

Figure 3. The grammatical relation hierarchy

For each grammatical relation, we defined one or more patterns over the phrase structure parse tree, produced by the tagger. Conceptually, each pattern is matched against every tree node, and the matching pattern with the most specific grammatical relation is taken as the type of the dependency. The patterns are based on English simple sentences, are domain independent and agree with the OntoNL Language model (figure 2).

Language Model. The system uses WordNet [Miller et alii, 1990] to obtain all the possible senses and synonyms of words in the user input. The linguistic analysis procedure concludes to a language model, a structure of Java classes. In this model diagram there are classes representing the grammatical relations that are connected with associations (figure 2).

2.2 The OntoNL Semantic Disambiguator

Disambiguation in natural language processing is used 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 is OWL Ontologies and instances of the language model produced by the Syntactic

Analyzer. 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:

1. The natural language expression contains general keywords that can be resolved by using only the ontology repository (ontological structures and semantics).
2. One of the subject or object part of the language model contains terms that cannot be disambiguated by using the ontology repository.
3. Neither the subject nor the object part contains terms disambiguated by using the ontological structures.

Figure 4 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.

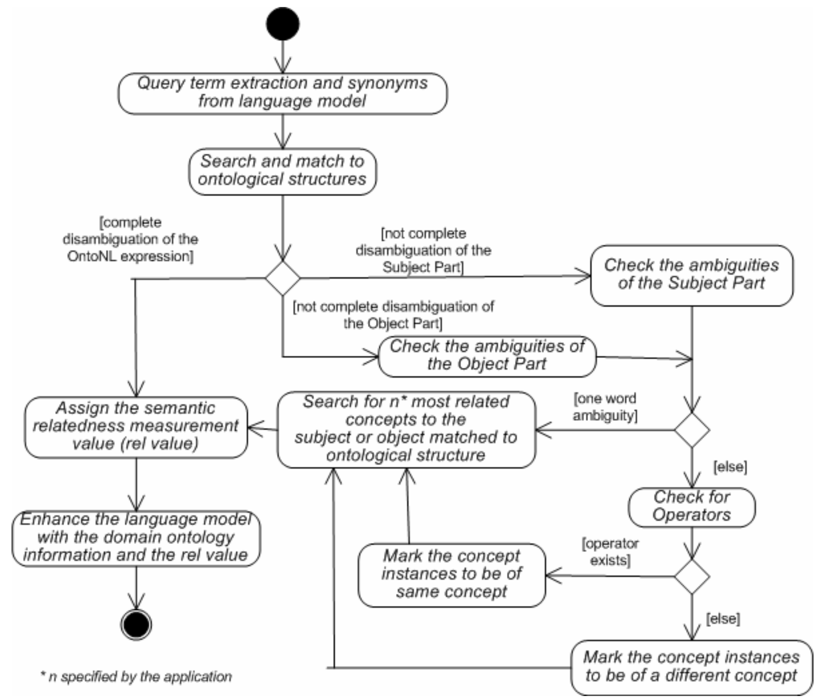

Figure 4. The OntoNL Semantic Disambiguation procedure

The input to the algorithm are instances of the language model, which include terms extracted from the natural language input, their synonyms, and their tagging according to the language model constructs. The algorithm searches to see if there is a correspondence between the naming of the language 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 (either in the Subject Part or the Object Part) the algorithm checks for the number of the terms that show ambiguity. If there is only one term with an ambiguity then the algorithm checks and retrieve 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 a part of the expression 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 Language Model with the corresponding ontology concepts to natural language terms in the class attribute and with the relatedness measurement value the value attribute.

2.3 The OntoNL Ontologies Processor

When a query cannot be disambiguated completely from the OntoNL Semantic Disambiguation procedure, OntoNL uses a Semantic Relatedness Measure 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. Intuitively, tightly interrelated concepts or clusters of concepts in the ontology are more likely to be the object of the user natural language interactions.

The relatedness measure depends on the semantic relations defined by properties in OWL. Properties can be used to state relationships between individuals (named *ObjectProperties*) or from individuals to data values (named *DatatypeProperties*). Based on the semantic relations when we detect that a source concept-class is immediately related via an *ObjectProperty* with the target concept, the relatedness value is set to 1

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 (how far is their common father) and the reciprocity of their properties (if the connecting OWL *ObjectProperties* have also inverse properties). The position of the concepts relatively to the position of their common subsumer will be examined by the conceptual distance and the specificity measurement.

We first count the number of the common properties of the two concepts. We then count the number of the common properties the two concepts share that are OWL:*inverseOf* properties:

$$rel_{prop}(c_1,c_2)=(f_1\times\frac{\sum_{i=1}^{n}p_{ijk}}{\sum_{i=1}^{n}p_{ij}})+(f_2\times\frac{\sum_{i=1}^{n}p_{invijk}}{\sum_{i=1}^{n}p_{ijk}}), \quad (1)$$

where $f_1 \geq f_2$ (2) and $f_1 + f_2 = 1$ (3).

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 two measures are combined with a weight that shows the relative importance of these two factors (f values).

The related senses measure counts the common senses of two concepts. It uses the sets of nouns for each concept that are synonyms and nouns extracted from the descriptive part of the glosses of the concept. Glosses are descriptions of a term meaning. 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:

$$rel_{RS}(c_1,c_2)=\frac{|S_1\cap S_2|}{|S_1\cap S_2|+|S_1\setminus S_2|}, \quad (4)$$

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

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 value of distance is calculated as:

$$pathDist(c_1,c_2)=\frac{d_{C_1}+d_{C_2}}{2*D}, \quad (5)$$

where d_1 is the number of edges from concept 1 to the closer common subsumer and d_2 the number of edges from concept 2 to the closer common subsumer. D is the maximum depth of the ontology. 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 path distance measure becomes

$$\forall d_{C1} \geq 1, d_{C2} \geq 1, d_{C1} + d_{C2} > 2 :$$
$$pathDist(c_1, c_2) = \frac{d_{C1} + d_{C2}}{2 * D} \in (0,1] \quad , (6)$$

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 class/subject of the language model, the two concepts are more related. When the direction of the path changes far from the reference class then the semantics change as well (more specialization). This calculation makes also the relatedness measure asymmetric, a desirable property in natural language processing applications.

As we have already stated when the value of d_{C1} is close to the value of $(d_{C1}+d_{C2})/2$ then the relatedness must be decreased, because the initial concept c1 is specialized a lot in comparison with the subsumer concept.

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

$$w1_{spec_{C1}} = \begin{cases} -\log \frac{2 \times d_{C1}}{d_{C1} + d_{C2}} \in (0,1], & \text{if } d_{C1} < \frac{d_{C1} + d_{C2}}{2} \\ 0, & \text{if } d_{C1} \geq \frac{d_{C1} + d_{C2}}{2} \end{cases} \quad , (7)$$

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

$$spec_{C1} = \frac{\#ObjP_{C1} - \#ObjP_S}{\#ObjP_S} \in [0, \infty) \quad (8)$$

were $ObjP_{C1}$ is the number of Object Properties of the concept C_1 and $ObjP_S$ is the number of ObjectProperties of the subsumer concept. If the factor becomes 1 or greater then the specialization is so big that we cannot count the relatedness based on the specificity. 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. We normalize the factor and we subtract it from 1, with the restriction that the number of the ObjectProperties of the concept – C_1 is at most 10 times the number of the ObjectProperties of the subsumer.

$$\forall \#ObjP_{C1} \leq 10 \times \#ObjP_S : w2_{spec_{C1}} = 1 - \log \frac{\#ObjP_{C1}}{\#ObjP_S} \in [0,1] \quad , (9)$$

else $w2_{spec_{C1}} = 0$ (10)

The conceptual distance measure then becomes

$$rel_{CD} = (w1_{specC1} + w2_{specC1} + 1 - pathDist(c_1, c_2)) / 3 \quad , (11)$$

The overall relatedness measure is the following:

$$\forall w_1 + w_2 + w_3 = 1, (w_1, w_2, w_3) > 0,$$
$$rel_{PROP}(c_1, c_2), rel_{CD}(c_1, c_2), rel_{RS}(c_1, c_2) \in [0,1] : \quad , (12)$$
$$rel_{OntoNL} = w_1 \times rel_{PROP} + w_2 \times rel_{CD} + w_3 \times rel_{RS} \quad , (13)$$

The three factors w_1, w_2 and w_3, help of balancing among the parameters depending on the application ontology. The parameters that we take into account for determining the factors' value are the language the ontology uses for its terminology, the number of the properties over the concepts and the depth of the domain ontology.

The measure is applied in all concepts of the ontology in the preprocessing phase and constructs 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.

2.4 The OntoNL Query Formulator

After the syntactic and semantic disambiguation, we have concluded to the subject of the query, specialized by additional description that forms the object part or possible object parts of the query. We need a formal way to represent the query, a standardized query language that will meet the specification of the ontology language (OWL) and will be easily mapped to various forms of repository constructions. Although we could in principle use an internal representation of the preprocessed NL interactions, we opted to use a representation that is near to the languages used in the Semantic Web, so that when the repository is based on OWL or RDF to be able to directly use it to access the repository. 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.

To provide an automatic construction of SPARQL queries we need at any point to define the path that leads from the subject part to the object part of the natural language expression by taking into account the constraints that are declared from the keywords and the relatedness value between the related classes of the ontology. The path connecting the classes directed from the user expression is given by an algorithm solving the problem:

Given a connected graph G = (V,E), a weight d:E->R+ and a fixed vertex s in V, find a optimized path from s to each vertex v in V. The optimized path is determined by the highest normalized sum value of the weights of the related concepts.

In the OntoNL Framework the edges linking the classes of the ontology graph are the objectProperties of the OWL syntax and the weight values are specified by the relatedness measure calculation described earlier in this chapter.

The general algorithm of the OntoNL query representation of domain-ontology disambiguated natural language expression in SPARQL is shown in figure 5:

```
  Program String SPARQLRepr (List, List, DoubleList)
   List subjOper, objOper, Values, OptPath;
   Double relVal;
   DoubleList SemRelMeas, ListNLStoOnto, ListNLOtoOnto;
    String Query, QueryTemplate, OntoSubjTerm, OntoObjTerm, value, value1, value2, val1,
 val2;
  Begin
  QueryTemplate="    PREFIX    ins:<ontology_path>    SELECT    ?OntoSubjTermIDs    WHERE
{?OntoSubjTermIDs rdf:type ?OntoSubjTerm ."
  If ListNLOtoOnto.size()=0 && subjOper.size()=0
   OntoSubjTerm = ListNLStoOnto.get(term)
   Query = QueryTemplate + "}";
  ElseIf ListNLOtoOnto.size()=0 && subjOper.size()!=0
   For all terms i of ListNLStoOnto
   OntoSubjTerm(i) = ListNLStoOnto.getTerm(i)
   Query = QueryTemplate + "}";
  Else
   relVal = ListNLOtoOnto.get(relatedness value)
   value = Values.get(not_Disambiguated_Term)
   If objOper.size()=0 && relVal=1
     OntoObjTerm = ListNLOtoOnto.get(term)
     Query = QueryTemplate +
     "{{?OntoSubjTerm ins:hasObjPropTo ?OntoObjTerm . "
     "?OntoObjTerm ins:hasDataProp "value"}"
   ElseIf objOper.size()=0 && relVal!=1
     OntoObjTerm = ListNLOtoOnto.get(term)
     OptPath = findOptPath(OntoSubjTerm, OntoObjTerm)
     Query = QueryTemplate + "
     For all ObjProperties of OptPath
     "{{?OntoSubjTerm ins:OptPath.get(hasObjProp) ?OntoObjTerm . "
     "?OntoObjTerm ins:hasDataProp "value"}"
   Else
     For all terms of OntoObjTerm
       OntoObjTerm = ListNLOtoOnto.get(term)
     If Values.size() = 1
       If relVal=1
         Query = QueryTemplate +
```

```
       "{{?OntoSubjTerm ins:hasObjPropTo ?OntoObjTerm1."
       "?OntoObjTerm1 ins:hasDataProp ?val1}UNION"
       "{{?OntoSubjTerm ins:hasObjPropo ?OntoObjTerm2."
       "?OntoObjTerm2 ins:hasDataProp ?val2}"
       "FILTER(?val1 = "value" || ?val2 = "value")"
     Else
       Query = QueryTemplate +
       For all ObjProperties of OptPath
       "{{?OntoSubjTerm ins:Opt.get(hasObjProp) ?First_Rel_Class ."
         "?First_Rel_Class ins:hasDataProp ?val1} UNION"
       "{{?OntoSubjTerm ins: OptPath.get(hasObjProp) ?Sec_Rel_Class."
       "?Sec_Rel_Class ins:hasDataProp ?val2}"
       "FILTER(?val1 = "value" || ?val2 = "value")"
   Else
     For all terms of Values
       If relVal=1
         Query = QueryTemplate +"
         "{{?OntoSubjTerm ins:hasObjPropTo ?First_Rel_Class."
         "?First_Rel_Class ins:hasDataProp "value1"}UNION"
       "{{?OntoSubjTerm ins:hasObjPropTo ?Sec_Rel_Class."
       "?Sec_Rel_Class ins:hasDataProp "value2"}"
     Else
       Query = QueryTemplate +"
       For all ObjProperties of OptiPath
         "{{?OntoSubjTerm ins: OptPath.get(hasObjProp) ?First_Rel_Class."
         "?First_Rel_Class ins:hasDataProp "value1"}UNION"
         "{{?OntoSubjTerm ins: OptPath.get(hasObjProp) ?Sec_Rel_Class."
         "?Sec_Rel_Class ins:hasDataProp "value2"}"
 End
```

Figure 5. The OntoNL query representation of domain-ontology disambiguated natural language expression in SPARQL

3 Overview of the NL2DL System

The NL2DL System is an application of the OntoNL Framework that addresses a semantic multimedia repository with digital audiovisual content of soccer events and metadata concerning soccer in general.

The overall architecture is shown in figure 6. The OntoNL expects domain ontology expressed in OWL. The reference ontologies we used is an application of the DS-MIRF ontological infrastructure [Tsinaraki *et alii* 2004] and the WordNet for the syntactic analysis. The repository for accessing the instances is the DS-MIRF Metadata Repository [Tsinaraki *et alii* 2006].

The OntoNL Component provides the NL Ontology API and the NL Query API for communication. The NL Query API contains functions to input a natural language query and after the disambiguation outputs a number of weighted SPARQL queries, based on the structures of the ontologies used for the disambiguation. It implements functions for the data transfer between the Framework and the repository. The NL Ontology API consists of the total of functions used for manipulating the ontologies that interfere with the system.

The DS-MIRF OntoNL Manager provides the OntoNL component with the ontologies for the disambiguation and the natural language expression for disambiguation. It is also responsible for retrieving the user request, communicate with the repository, manage the results, rank them based on any existing User Profile information and presented them to the front end the user uses for interactions.

The output of the OntoNL is weighted SPARQL queries. To interface with DS-MIRF we had to develop mappings of the SPARQL to the retrieval language of DS-MIRF which intern uses XQuery to access semantic MPEG-7 multimedia content from the XML DBMS.

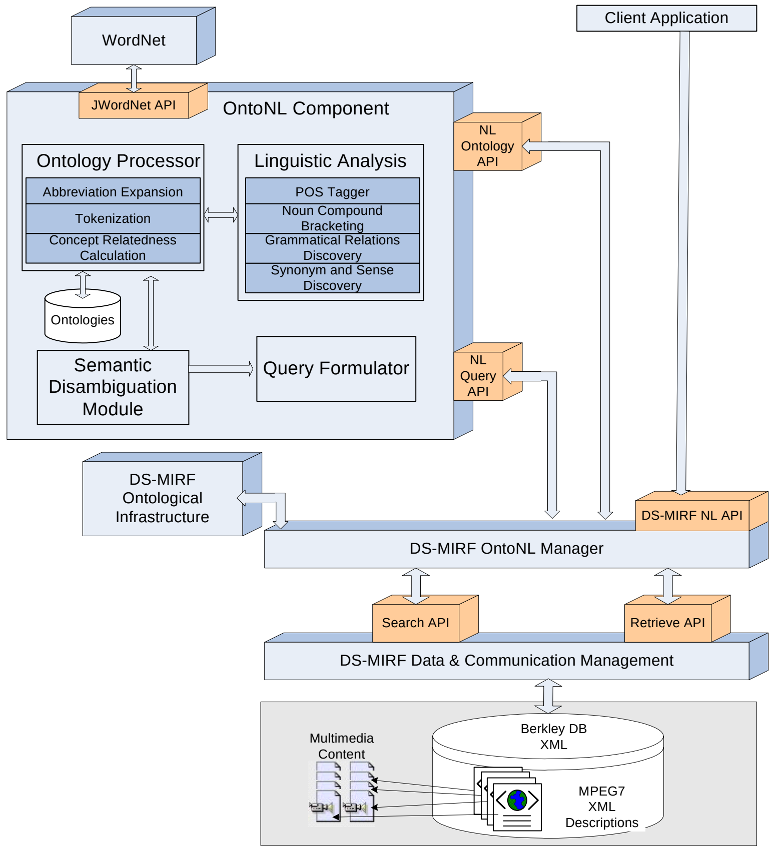

Figure 6. NL2DL Infrastructure

4 Experimental Evaluation

A complete evaluation framework has been designed for the OntoNL generator. A screenshot of the NL2DL system for retrieving semantically indexed audio visual content in the domain of the FIFA World Cup 2006 can be seen in figure 7. The application also includes the option of inserting the request using speech, but this is not described in this paper. The result view presents a list with the labels of the XML descriptions that comprise the requested information. The user can choose to see the audiovisual content of the results.

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 two objects. Budanitsky and Hirst [2006] based on their study have found that comparing WordNet similarity measures with human judgments give the best assessments of the “goodness” of a measure.

After this step we continued with an application-based evaluation of the OntoNL measure. We chose to use for the application, the OWL Ontology for the domain of soccer (http://lamia.ced.tuc.gr/ontologies/AV_MDS03/soccer), because it is a big and very specific ontology. Also, the context of the ontology is familiar with the users.

The next sub-section will describe the experimental setup and the statistical evaluation of the results setting the stage for a discussion of the results.

Figure 7. The multimedia object retrieval by the NL2DL System

4. 1 Study Design

Experiments that rely on human judgments have become the benchmark in determining the similarity of words in NLP research [Budanitsky and Hirst, 2006], [Jiang and Conrath, 1997], [Leacock and Chodorow, 1998], [Rada, and Bicknell, 1998], [Rada *et. alii*. 1989.]. 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 found a number of suitable object pairs from a number of ontologies freely available in the web. The OWL domain ontologies have a number of advantages. Each class has a variety of relationships to attributes, exceptions, etc. and in many cases a detailed textual description is provided.

Then, we defined an appropriate order in which those pairs were going to be presented to the subjects. We point out that the subjects' ability to relate to the ontology content (domain) was crucial for the success of the experiment. From the ontologies we have selected a number of concepts that we thought would be understandable to a general audience and combined them into pairs 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 have obtained relatedness judgments from 20 human subjects, 10 from the computer science field that had some knowledge of the domain ontologies and 10 from the liberal arts field, that were used for the evaluation, for 85 pairs of concepts that we meet in seven OWL domain ontologies [Ontologies used for experimentation] freely available on the web, for different domains. The main goal was to compare the OntoNL sub-measures 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 [Gentner and Medina, 1998], but also useful to us for the determination of the impact of the sub-measures of relatedness to the overall measure. Finally, the subjects could add some comments on their assessment.

4. 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. 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 Table1.

We observe that w_1 and f_1 are in general the most important of the weights, which implies that the number of common 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 wine ontology) 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 five ontologies, while the conceptual distance measure (w_2) had a stronger impact in two ontologies.

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 each one of the ontologies used for the specific experimentation.

Ontology	rel_{PROP}		rel_{OntoNL}		
	f_1	f_2	w_1	w_2	w_3
Soccer Ontology	0,5	0,5	0,7	0,1	0,2
Wine Ontology	0,8	0,2	0,3	0,2	0,5
People Ontology	0,8	0,2	0,45	0,2	0,35
Pizza Ontology	0,9	0,1	0,65	0,1	0,25
Koala Ontology	0,9	0,1	0,3	0,5	0,2
Images Ontology	0,7	0,3	0,6	0,3	0,1
Travel Ontology	0,9	0,1	0,7	0,1	0,2

Using the optimal values for the parameters we studied how the computed relatedness measure among two concepts was correlated with the relatedness perceived by the human subjects. Table 2 shows the computed correlation coefficients between the system computed relatedness measure and the human subjects evaluated relatedness for two classes of human subjects, one coming from the human liberal arts field and one coming from the computer science field.

Table 2: The values of the coefficients of correlation between human ratings of relatedness and four computational measures; the three submeasures that constitute the OntoNL Semantic Relatedness Measure and the overall OntoNL measure with relative weights of Table 1

Humans LibArts Field				
Measure	rel_{PROP}	rel_{CD}	rel_{RS}	rel_{OntoNL}
Soccer Ontology	0,964	0,935	0,938	0,978
Wine Ontology	0,948	0,947	0,967	0,981
People Ontology	0,954	0,927	0,953	0,969
Pizza Ontology	0,874	0,854	0,832	0,927
Koala Ontology	0,898	0,921	0,892	0,936
Images Ontology	0,927	0,914	0,911	0,954
Travel Ontology	0,958	0,924	0,932	0,970
Humans CompSc Field				
Measure	rel_{PROP}	rel_{CD}	rel_{RS}	rel_{OntoNL}
Soccer Ontology	0,975	0,929	0,908	0,988
Wine Ontology	0,943	0,961	0,925	0,972
People Ontology	0,963	0,982	0,917	0,987
Pizza Ontology	0,917	0,903	0,899	0,921
Koala Ontology	0,914	0,932	0,911	0,938
Images Ontology	0,922	0,932	0,906	0,962
Travel Ontology	0,954	0,922	0,923	0,963

The results are satisfactory and show that the average correlations for each ontology were always more than .9 and in 10 out of the 14 cases they were more than .95. The average correlation was .96.

The human subjects also evaluated the relatedness of the concepts based on the semantic measure that we have developed (common properties, related senses, and conceptual distance). The correlations of their evaluations with the system computed measures are shown in Table 2, and are also satisfactory. It is interesting to observe that the subjects with computer science background had higher correlations with the system for the conceptual distance measure, while human subjects from liberal arts had higher correlations in general for the related properties measure. 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).

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, 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.

Table 3: 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.

OWL Domain Ontologies	rel_{PROP}		rel_{OntoNL}		
	f_1	f_2	w_1	w_2	w_3
	0,8	0,2	0,6	0,17	0,23

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 (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 correlations however between human subject and the system evaluations, are quite high. The average drop in correlation was -02 (from .96 to .94), while the maximum drop in some ontologies was .04. In only one case (Koala Ontology) the average correlation dropped below .9 (to .892). For this ontology however, even with its optimal weights the correlation was not very high (.936).

Table 4: The values of the coefficients of correlation between human ratings of relatedness and four computational measures; the three submeasures that constitute the OntoNL Semantic Relatedness Measure and the overall OntoNL measure with relative weights of Table 3

Humans LibArts Field		
Measure	rel_{OntoNL}	rel_{OntoNL}’
Soccer Ontology	0,978	0,940
Wine Ontology	0,981	0,953
People Ontology	0,969	0,952
Pizza Ontology	0,927	0,907
Koala Ontology	0,936	0,892
Images Ontology	0,954	0,953
Travel Ontology	0,970	0,962
Humans CompSc Field		
Measure	rel_{OntoNL}	rel_{OntoNL}’
Soccer Ontology	0,988	0,946
Wine Ontology	0,972	0,948
People Ontology	0,987	0,954
Pizza Ontology	0,921	0,923
Koala Ontology	0,938	0,911
Images Ontology	0,962	0,960
Travel Ontology	0,963	0,948

We plan to pursue more research in the future, using more ontologies, for determining automatically the weights for any given ontology. Parameters that we have found that affect the choice of weight are the following:

The language the ontology uses for its terminology. When ontologies are used directly from their source (web) a major factor of the relRS parameter’s performance is the names that are used to describe the ontologies. If the names for the concepts and the logical relationships among the concepts used are near the “natural language” names the performance of the system is significantly better.

The number of the properties over the concepts. When the concepts of the ontology have a number of properties that specialize them over other concepts (the semantic network has a significantly greater number of edges over nodes) then the parameter relPROP can participate with a great value of influence in the overall OntoNL semantic relatedness measure calculation.

The depth of the domain ontology. When the ontology is of a great depth then the conceptual distance needs to be assigned with a big relative weight because the information loss is significant over the inheritance.

The overall performance of the natural language interfaces from the point of view of the satisfaction of users with the interaction with a natural language interface is being evaluated using human subjects for the domain of soccer videos. Preliminary results indicate that the users are considering the natural language interface generated by the OntoNL Framework to a video repository successful. A voice recognizer has been integrated in the soccer application and a much more extensive system evaluation of the complete system will be pursued.

As far as it concerns the application-based evaluation, the experiments tested if the language model’s components where successfully mapped to ontological structures (figure 8) and if the semantic relatedness measure resulted in satisfactory matches (figure 9). We also present the overall satisfaction of users with respect to the effectiveness of the results compared against a keyword-based search (figure 10). The conclusion that that derives is that in a second iteration of tests the users expressed a higher satisfaction because their familiarity of using the system increased. The results that concern ontological structures and semantics (figures 8 and 9) are strongly dependent on the form of the specific ontology. Overall, the performance decreases a little as the complexity of the language model increases, but as shown in figure 10, we get the correct results sooner and faster against a keyword-based search.

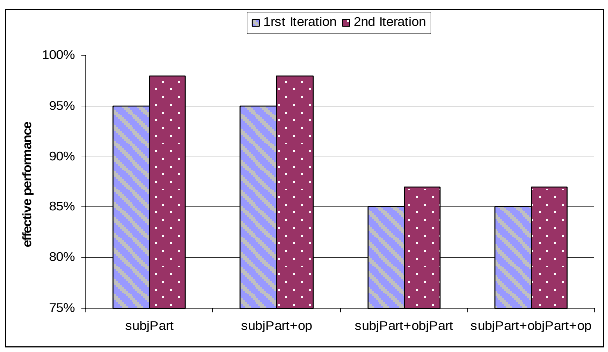

Figure 8. The effectiveness of ontology mappings (DS-MIRF ontologies for the domain of soccer) to user input

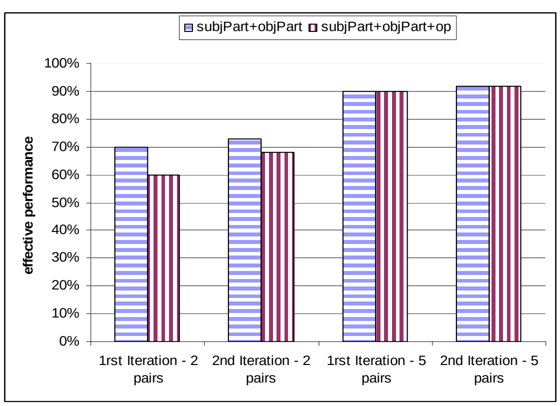

Figure 9. The effectiveness of the semantic relatedness measure in the DS-MIRF ontologies for the domain of soccer

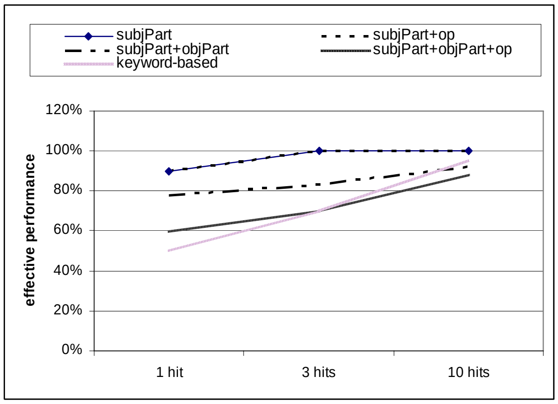

Figure 10. The effectiveness of the NL2DL in the domain of soccer against a keyword-based search

5. Related Work

The NLP literature provides the largest group of related work in the semantic relatedness measurement field. The known methodologies for measuring semantic relatedness are based on lexical resources or WordNet [Fellbaum 1998] and other semantic networks or deal with computing taxonomic path length. All approaches that we are aware of measuring semantic relatedness that use a lexical resource construe the resource, in one way or another, as a network or directed graph, and then base the measure of relatedness on properties of paths in this graph [Leacock and Chodorow 1998], [Lin 1998].

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 [Resnik 1990]. 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 alii*, [1989], [1998]. The principal assumption of Rada and colleagues was that “the number of edges between two terms in the MeSH hierarchy is a measure of conceptual distance between the terms”.

Jiang and Conrath [1997] 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 [2006] support our claim that the quality of similarity measures is dependent on the ontology in general. They find that differences in the quality of WordNet-based similarity measurement algorithms found in various papers can be explained by the different versions of WordNet that have been used. To confront with this issue Lin [1998] 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 [Resnik 1990]. The drawback is that it still requires a probabilistic model of the application domain, retrieved by parsing a large word corpus. This limitation makes it problematic for smaller ontologies.

All the research results presented in the literature so far [Budanitsky and Hirst 2006], [Jarmasz and Szpakowicz 2003], [Jiang and Conrath 1997], [Leacock and Chodorow 1998], [Lin 1998], [Rada *et alii* 1989], [Rada and Bicknell 1998], [Resnik 1990] and [Wu and Palmer 1994] were tested on specific ontologies like the WordNet and the 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 and not relatedness between concepts. Also, most of these approaches are focused on the comparison of nouns, limiting their generality to complex objects or even hierarchies of verbs.

As far as it concerns the natural language interfaces to knowledge repositories in general, the natural language interaction research has very recently started to explore semantic disambiguation using ontologies, providing systems like AQUA [Vargas-Vera *et alii* 2004). AQUA is a question-answering system which amalgamates Natural Language Processing (NLP), Logic, Ontologies and

Information Retrieval techniques to provide answers to queries in a specific domain. AQUA translates English questions into logical queries that are then used to generate of proofs and is coupled with the AKT reference ontology for the academic domain, written in OCML. The system works in a pattern-matching mode trying to find exact matches with names in the ontology and hasn't been tested in other domains using other ontologies.

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 [Karanastasi *et alii* 2004]. Ontologies used for this system were capturing the TV-Anytime standard (http://www.tv-anytime.org). The system used both ontologies and User Profiles in order to do semantic natural language processing. Keywords described the domain ontologies and user profile information was also used for disambiguation with satisfactory results. The approach was tuned and depended on the specific ontologies and lacked the generality and the completeness of the system described here. In addition the domain ontologies were based on keywords and not on deep knowledge structures.

There is a significant progress in natural language processing research and especially in the computational linguistics area. Numerous approaches exist for automatic assignment of parts of speech, that use top performing methods, such as Hidden Markov Models, maximum entropy approaches [Ratnaparkhi 1996] and transformation-based learning. The maximum entropy approach has been adopted in [Toutanova *et alii* 2003] and concluded in the Stanford LogLinear Part of Speech-Tagger (http://nlp.stanford.edu/software/tagger.shtml) that gave 96.86% accuracy on the Penn Treebank. Noun compound bracketing is a major field of interest in the natural language processing procedure. Different models like the dependency [Layer 1995] and the adjacency model [Marcus 1980] have been developed with quite accurate results. Systems for extracting grammatical relations from English sentences have been developed [De Marneffe *et alii* 2006] with good results but based on practical rather than theoretical concerns for modeling grammatical relation schemas [Carroll *et alii* 1999].

In comparison with natural language interfaces that focus either on developing methodologies only for syntactic analysis or for a specific application, the OntoNL Framework is able to address uniformly a range of problems in sentence analysis each of which traditionally had required a separate computational mechanism. In particular a single architecture handles both syntactic and semantic ambiguities, handles ambiguity at both a general and a domain specific environment and uses semantic relatedness measures on the concepts of the ontology to provide better ranked results.

6. Conclusions

We have presented the OntoNL software engineering Framework for the generation of natural language user interfaces to knowledge repositories. The framework contains syntactic and semantic analysis components which are supported by a language model. The semantic analyzer utilizes domain ontologies described in OWL to try to disambiguate or rank the possible user queries.

We have presented the OntoNL Ontology-driven Semantic Relatedness measure for OWL ontologies used for natural language disambiguation. 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, their specificity in comparison with their common root concept and the semantic relations to other ontological concepts. 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 a measure that has a great influence if the ontology depth is big since in that case there are 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 results of the experiments with 7 OWL domain ontologies, freely available on the web [Ontologies used for experimentation] are presented as a comparison of the measurement of relatedness between human subjects and the OntoNL measure.

We have observed that when ontologies are used directly from their source (web) a major factor in the performance of the natural language interaction system is the names that are used to describe the ontologies. This may imply that for ontologies that do not utilize "natural language" names for their concepts and relationships we have to provide a mapping to more natural language expressed ontologies.).

Overall, we found that the semantic relatedness measure that is used for the ontology-based semantic ranking of concepts for natural language disambiguation is quite complete and shows very good results. For future improvements, we may need to investigate the influence of more complex structures of OWL vocabulary to the performance.

After this step we continued with an application-based evaluation of the OntoNL measure. We chose to use for the application, the OWL Ontology for the domain of soccer (http://lamia.ced.tuc.gr/ontologies/AV_MDS03/soccer), because it is a big and very specific ontology. Also, the context of the ontology is familiar with the users. The results show that the implementation of semantic natural language interactions with semantic repositories is feasible and inexpensive for a large number of applications domains and domain ontologies.

References

Budanitsky, A., Hirst, G. 2006 Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics 32*(1): 13-47

Gentner D., Medina, H. 1998 Similarity and the Development of Rules, *Cognition* 65: 263-297

Jarmasz, M., Szpakowicz, S. Roget's Thesaurus and semantic similarity. 2003. *In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-2003)*, pp. 212-129

Jiang, J. J., Conrath, D. W. 1997. Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. *In Proceedings of International Conference on Research in Computational Linguistics*

Karanastasi, A., Kazasis, F., Christodoulakis, S. 2004. A Natural Language Model for Managing TV-Anytime Information from Mobile Devices. *In Proceedings of the NLDB 2004*

Kozima, H., Furugori, T. 1993. Similarity between words computed by spreading activation on an English dictionary. *In Proceedings of 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL-93)*, pages 232–239, Utrecht

Lauer, M. 1995 *Designing Statistical Language Learners: Experiments on Noun Compounds.* Ph.D. thesis, Department of Computing Macquarie University NSW 2109 Australia

Leacock, C., Chodorow, M. 1998. Combining local context and WordNet similarity for word sense identification. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database,* chapter 11, pages 265–283. The MIT Press, Cambridge, MA.

Lin, D. 1998 An information-theoretic definition of similarity. *In Proceedings of the 15th International Conference on Machine Learning*

Marcus, M. 1980. *A Theory of Syntactic Recognition for Natural Language*. MIT Press.

Marneffe, de, M. C., MacCartney, B., Manning, C. D., 2006, Generating Typed Dependency Parses from Phrase Structure Parses, *In Proceedings of the LREC 2006*

Miller, G., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K.J. 1990. Introduction to WordNet: an on-line lexical database. *Int. Jrnl of Lexicography,* 3(4), 235 – 244

Ontologies used for experimentation:
Wine: http://www.w3.org/TR/owl-guide/wine.rdf
Pizza Ontology: http://www.co-ode.org/ontologies/pizza/2005/05/16/pizza.owl
Koala Ontology: http://protege.stanford.edu/plugins/owl/owl-library/koala.owl
People Ontology: http://owl.man.ac.uk/2005/07/sssw/people.html
Images Ontology: http://www.mindswap.org/~glapizco/technical.owl
Travel Ontology: http://learn.tsinghua.edu.cn/homepage/2003214945/travelontology.owl
Soccer Ontology: http://lamia.ced.tuc.gr/ontologies/AV_MDS03/soccer

Rada, R., Bicknell, E. 1998. Ranking documents with a thesaurus. *JASIS,* 40(5):304–310, September

Rada, R., Mili, H., Bicknell, E., Blettner, M. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics,* 19(1):17–30, February.

Ratnaparkhi, A. 1996 A Maximum Entropy Model for Part-of-Speech Tagging. *In Proceedings of EMNLP 1996*

Resnik, P. Using information content to evaluate semantic similarity. 1995. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence,* pages 448–453, Montreal, Canada

Toutanova, K., Klein, D., Manning, C. D., Singer, Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network, *In Proceedings of Human Language Technologies (HLT) 2003*

Tsinaraki C., Polydoros P., Christodoulakis S., 2004. Integration of OWL ontologies in MPEG-7 and TVAnytime compliant Semantic Indexing, *In Proceedings of the CAiSE 2004*

Tsinaraki C., Christodoulakis S. 2006. A User Preference Model and a Query Language that allow Semantic Retrieval and Filtering of Multimedia Content. *In Proceedings of SMAP 2006*

Vargas-Vera, M., Motta, E. 2004. AQUA – Ontology-based Question Answering System. *In Proceedings of the MICAI 2004*