

Ontology-Driven Semantic Ranking for Natural Language Disambiguation in the OntoNL Framework

Anastasia Karanastasi and Stavros Christodoulakis

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

Abstract. The measurement of the semantic relatedness has many applications in natural language processing, and many different measures have been proposed. Most of these measures use WordNet as their central resource and not domain ontologies of a particular context. 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, a natural language interface generator for knowledge repositories. The experimentations show a good performance in a number of OWL ontologies.

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. 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 from the OWL ontology that was used as a reference schema for constructing a knowledge repository.

This methodology is integrated in the OntoNL Framework [3], a natural language interface generator to knowledge repositories. 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. The communication is done through APIs. Note that different domain ontologies may be just imported in the system, provided that they are expressed in the same knowledge representation language (OWL). The Framework is therefore reusable with different domain ontologies.

We examine how consulting domain ontologies can help to do semantic language processing and disambiguation, not just syntactic. To this end, we have developed and evaluated a semantic relatedness measure 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.

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, 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/>).

2 Related Work

The known methodologies for measuring semantic relatedness are based on lexical resources or WordNet [2] and other semantic networks or 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 [4], [5].

Most of the methods use the WordNet [1], a broad coverage lexical network of English words, as a semantic network. Nouns, verbs, adjectives, and adverbs are

organized into synonym sets (synsets), each representing one underlying lexical concept, that are interlinked with a variety of relations. A simple way to compute semantic relatedness in a taxonomy such as WordNet is to view it as a graph and identify relatedness with path length between the concepts [9]. This approach was followed in other networks also, like the MeSH (Medical Subject Headings) (<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., [7], [8]. 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”.

Despite its apparent simplicity, a widely acknowledged problem with the edge-counting approach is that it typically “relies on the notion that links in the taxonomy represent uniform distances”, which is typically not true. Sussna’s approach to scaling [10], Wu and Palmer’s Conceptual Similarity [11] and Leacock and Chodorow’s normalized path length [5] are efforts in WordNet to overcome the problem of the edge-counting approach.

One last approach for measuring semantic relatedness attempts to counter problems inherent in the structures of a general ontology by incorporating an additional, and qualitatively different, knowledge source, namely information from a corpus [1], as was first proposed in [9]. The key idea underlying Resnik’s approach [9] is the intuition that one criterion of similarity between two concepts is “the extent to which they share information in common”, which in an IS-A taxonomy can be determined by inspecting the relative position of the most-specific concept that subsumes them both. In order to overcome the information loss in Resnik’s method, Lin presented a universal similarity measure [6]. Noticing that all of the similarity measures known to him were tied to a particular application, domain, or resource, Lin attempted to define a measure of similarity that would be both universal (applicable to arbitrary objects and “*not presuming any form of knowledge representation*”) and theoretically justified (“*derived from a set of assumptions*”, instead of “*directly by a formula*”, so that “*if the assumptions are deemed reasonable, the similarity measure necessarily follows*”). Lin’s measure also referred to similarities and he took into account only the commonality and differences of two terms. The objective was to compute the similarity of ordinal values and words.

All the research results presented in the literature so far [5], [6], [7], [9], [10], [11] were tested in specific ontologies like the WordNet and the MeSH ontology, 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.

In a framework like the OntoNL that needs to preserve its generality we could not rely on a general hierarchy of terms like the WordNet to disambiguate user expressions or the MeSH ontology a semantic hierarchy of terms used for indexing articles in the medical domain. 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 here 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 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. 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.

An important issue that we also attack is the need of an asymmetric measure, since all the previous approaches are based on symmetric measures. Asymmetric relatedness denotes that the relatedness between A and B is not necessarily the same as the relatedness between B and A. This is an important aspect for natural language processing since relations that are described with natural language do not indicate mathematical rules. Also, in a domain ontology we need to take into account the total of information loss in an IS-A taxonomy between the nodes that we want to test their similarity or relatedness and their common subsumer (common root node). The semantic information each concept inherits from the root node may be the same but its specialization defined by new properties that it carries is not the same for all the concepts.

All these parameters modulated the proposed semantic relatedness measure described in Section 4. We also need to point that this measure was developed to help the natural language disambiguation process when the use of domain ontologies is not enough to determine the sense words are used in an utterance for a specific domain of context, as it is described in Section 3.

3 The OntoNL Semantic Disambiguation Algorithm

The purpose of semantic disambiguation in natural language processing, 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 (WordNet gives only generic categories of senses and not domain specific. This domain specific disambiguation is much more powerful).

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 cannot be disambiguated by using the ontology repository.
3. Neither the subject nor the object part contains terms disambiguated using the ontological structures.

Next, we describe the entire semantic disambiguation algorithm based on the different levels of ambiguities using a UML Activity Diagram (Fig. 1). It is a general approach where the disambiguation is based on an OWL repository.

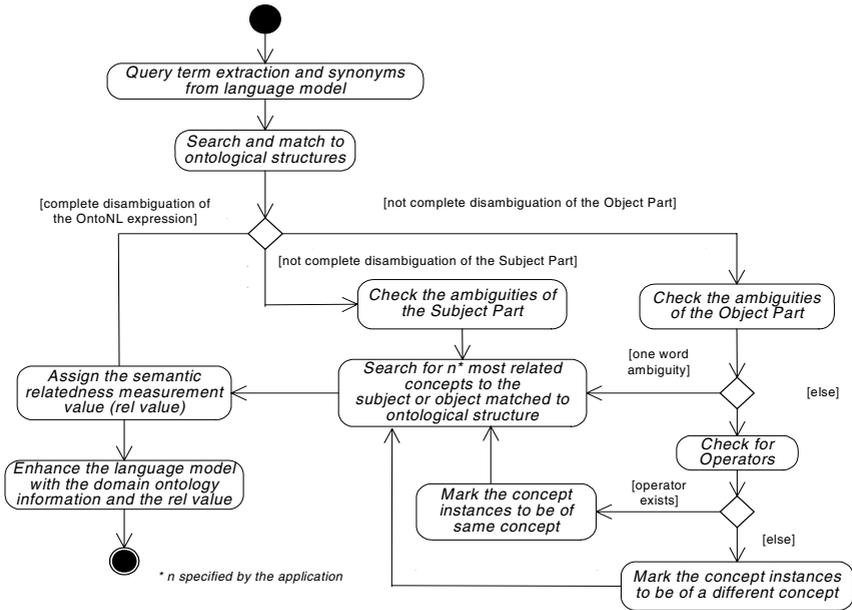


Fig. 1. The OntoNL Semantic Disambiguation procedure

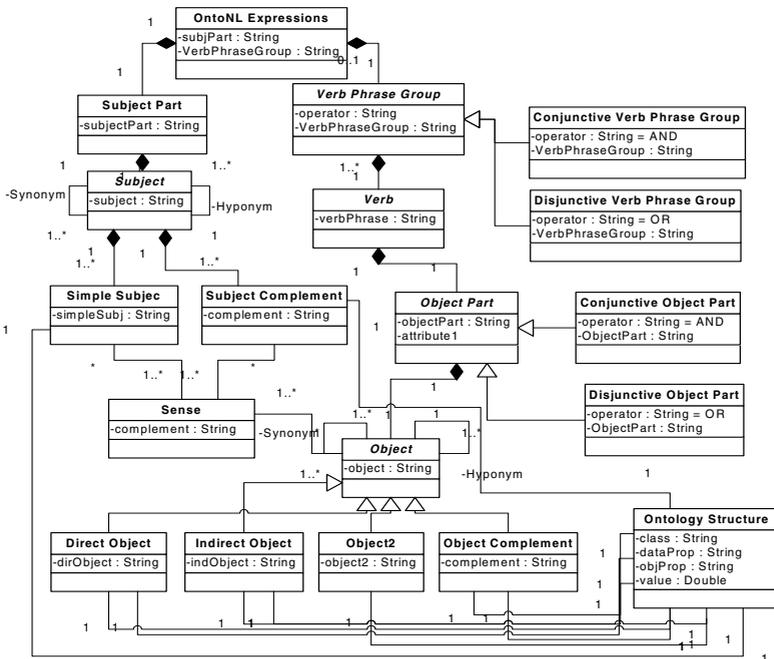


Fig. 2. The OntoNL Language Model that derives from the syntactic and semantic analysis, based on the OntoNL Natural Language Expressions

The language model that is referred in the last activity of the semantic disambiguation procedure of Fig. 1 is described by the UML Class Diagram of Fig. 2. In this model diagram there are classes representing the grammatical relations that are connected with associations. 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. 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 classes of the OntoNL language model are enhanced with the parsed information from the OntoNL syntactic disambiguation phase described in [3].

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 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 Language 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 Ontology-Driven Semantic Ranking

When a query cannot be disambiguated completely from the OntoNL Semantic Disambiguation procedure, OntoNL returns all the possible results ranked according to a value computed by the system that represents the possibility that the user has requested them. To compute the ranking of possible results, OntoNL borrows ideas and develops new ones from the research of Semantic Relatedness of concepts in a semantic network.

The output of the measurement of the relatedness between concepts of domain ontologies is a matrix containing a weight of relatedness between any two concepts. It is crucial to identify more specific domains inside the domain, based on concepts and relationships of those concepts. Consider an n^{th} row in this matrix and a function $F_n(i)$

which takes the n^{th} row and returns the set of the largest values. Then this function defines a local association cluster around the concept C_n . The clustering has the effect of reducing the size of a domain by creating groups of more specific information from one or more ontologies to search for semantic information.

The relatedness is also a metric that 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 related via an **ObjectProperty** with the target concept, the relatedness value is 1 independently from their commonality or common senses or conceptual distance.

The algorithm also takes into account the semantic relation of **EquivalentClass**. The EquivalentClass of the source class has a **similarity (not relatedness) value 1** with the source class in order to also consider the relatedness measurement value of the equivalent class with the remaining classes of the Ontology.

The **commonality** depends on the amount of the common information two concepts share. We cannot use commonality like it was used by Resnik [9] when we consider domain ontologies other than WordNet because there are no senses to count the frequency of a word. We can accept partly that the distance from the most specific common subsumer of the two concepts is a criterion that must be taken into account but we have also to consider the number of common relations. We do need to keep the measure asymmetric so it will depend on the reference concept of which the relatedness to another concept we calculate. The measure that we developed has two factors: The position of the concepts relatively to the position of their most specific common subsumer (how far is their common root node) and the relativity of their properties (OWL ObjectProperties):

To measure the relativity of the properties of any two concepts we first count the number of the common properties that the two concepts share.

$$rel_{c_1}(c_1, c_2) = \frac{\sum_{i=1}^n p_{i12}}{\sum_{i=1}^n p_{i1}} \quad (1)$$

The value p_{ij} represents the fact that concept c_j is related to concept c_i . The value p_{i12} represents the fact that both concepts c_1 and c_2 are related to concept c_i . This measure takes into account that concepts share more common properties with other concepts that relate.

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

$$rel_{c_2}(c_1, c_2) = \frac{\sum_{i=1}^n p_{invi12}}{\sum_{i=1}^n p_{i12}} \quad (2)$$

where the p_{invi12} represents the fact that both concepts are inversely related.

The motivation to measure the common **inverseOf** properties is to release the relatedness measure from the similarity dimension. If we only counted the common **ObjectProperties** then we would assign a great value of relatedness between siblings (subclasses with common superclass) which are similar but not semantically related as the OntoNL Framework defines.

The measures rel_{C_1} and rel_{C_2} are combined with relative weights that show the relative importance of these two factors (f values):

$$\forall f_1 \geq f_2, f_2 > 0, f_1 + f_2 = 1:$$

$$rel_{prop}(c_1, c_2) = (f_1 \times \frac{\sum_{i=1}^n P_{ijk}}{\sum_{i=1}^n P_{ij}}) + (f_2 \times \frac{\sum_{i=1}^n P_{inijk}}{\sum_{i=1}^n P_{ijk}}) \quad (3)$$

The factors f_1 and f_2 in general depend on the ontologies used, and we assume that they are experimentally determined for a given ontology. A systematic algorithm for the quantification of the factors is ongoing.

The **conceptual distance measure** is based on two factors; the path distance and the specificity. The **specificity** of the concepts is based on their position in the ontology (the leaf nodes are the most specific concepts in the hierarchy). The path distance counts the edges in the minimal path of edges from a concept to another. Within one conceptual domain, the relatedness of a concept (C_1) to another concept (C_2) is defined by how closely they are related in the hierarchy, i.e., their structural relations (IS-A relation). In the OntoNL, the IS-A relations are implemented through the `rdfs:subClassOf` syntax of OWL. The parameter that differentiates our measure from the classic measures of distance counting is the change of direction that is combined with the specificity factor. We claim that when the change of direction (from superclassing to subclassing) is close to the initial concept- c_1 (that is the **subject** of the natural language expression) of the pair we test the relatedness; the two concepts are more related. When the direction of the path changes far from the first concept then the semantics change quite as well (more specialization). Also we take into account the place of the concepts in the hierarchy. The terms located higher in the hierarchy have higher values of relatedness than located terms lower in the hierarchy.

The value of distance can be measured with the following measure

$$pathDist(c_1, c_2) = \frac{d_{C_1} + d_{C_2}}{2 * D} \in (0, 1] \quad (4)$$

where d_{C_1} is the number of edges to go from the concept 1 to the closer common superconcept (subsumer) and d_2 the number of edges to go from the concept 2 to the closer common superconcept (subsumer). With D we count 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 $d_{C_1} = 0$ and $d_{C_1} + d_{C_2} = 2$. So, the path distance measure becomes

$$\forall d_{C_1} \geq 1, d_{C_2} \geq 1, d_{C_1} + d_{C_2} > 2: pathDist(c_1, c_2) = \frac{d_{C_1} + d_{C_2}}{2 * D} \in (0, 1] \quad (5)$$

We need a factor to determine the specificity of the concepts inside the ontology. As we have already stated is the value of d_{c_1} is close to the value of $(d_{c_1}+d_{c_2})/2$ then the relatedness must be decreased, because the initial concept c_1 is specialized a lot in comparison with the subsumer concept.

$$w_{spec_{c_1}}^1 = \begin{cases} -\log \frac{2 \times d_1}{d_1 + d_2} \in (0, 1], & \text{if } d_1 < \frac{d_1 + d_2}{2} \\ 0, & \text{if } d_1 \geq \frac{d_1 + d_2}{2} \end{cases} \quad (6)$$

We, also use a method of counting the specialization of the concept – c_1 based on the object properties of the subsumer (root OWL Class), by the factor:

$$spec_{c_1} = \frac{\#ObjP_{c_1} - \#ObjP_s}{\#ObjP_s} \in [0, \infty). \quad (7)$$

were $ObjP_{c_1}$ 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 $spec_{c_1}$ 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.

$$\forall \#ObjP_{c_1} \leq 10 \times \#ObjP_s : w_{spec_{c_1}}^2 = 1 - \log \frac{\#ObjP_{c_1}}{\#ObjP_s} \in [0, 1]. \quad (8)$$

The conceptual distance measure then becomes

$$rel_{CD} = (w_{spec_{c_1}}^1 + w_{spec_{c_1}}^2 + 1 - pathDist(c_1, c_2)) / 3. \quad (9)$$

The **amount of related senses measure** is a measure that concerns the domain ontology and the WordNet Ontology. From the WordNet Ontology we exploit the noun glosses. Glosses are descriptions of a word's sense and it consists of a descriptive part and an example of use case. From the domain ontologies we exploit the concept descriptions that are expressed in the `<owl:label>` and `<owl:comment>` constructs. The measure is based on sets of each concept that contain synonyms and nouns extracted from the descriptive part of the glosses of each concept:

$$rel_{RS}(c_1, c_2) = \frac{|S_1 \cap S_2|}{|S_1 \cap S_2| + |S_1 \setminus S_2|}. \quad (10)$$

were S_1 is the description set of senses for concept C_1 and S_2 the description set of senses for concept C_2 .

The overall relatedness measure is the following:

$$\begin{aligned} \forall w_1 + w_2 + w_3 = 1, (w_1, w_2, w_3) > 0, \\ rel_{PROP}(c_1, c_2), rel_{CD}(c_1, c_2), rel_{RS}(c_1, c_2) \in [0, 1]: \\ rel_{OntoNL} = w_1 \times rel_{PROP} + w_2 \times rel_{CD} + w_3 \times rel_{RS} . \end{aligned} \quad (11)$$

The three factors w_1 , w_2 and w_3 , can help of choosing which parameter can better express the semantic relatedness. We test the values of the factors in the evaluation section.

The measure is applied in all concepts of the ontology in the preprocessing phase and constructs a $N \times N$ matrix, where N is the total number of concepts, with the relatedness values of each concept with all the other concepts inside the disambiguation ontology.

5 Representation of Natural Language Interactions

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 $d: E \rightarrow R^+$ 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 Fig. 3:

```

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:hasObjPropTo ?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

```

Fig. 3. The OntoNL query representation of domain-ontology disambiguated natural language expression in SPARQL

6 Evaluation

A complete evaluation framework has been designed. Such a framework takes into account a large number of parameters regarding the characteristics of the ontologies involved and the types of users.

Our objective so far was to integrate all the components involved, to test interoperability, to integrate with a knowledge repository and to experiment with the performance of the disambiguation component. The integration among the components is complete and serves all the needs currently anticipated. A complete scenario utilizing all the components of the system with semantic MPEG-7 descriptions of soccer games which utilize an extensive soccer ontology [10] was tested successfully.

We have focused now our attention to the performance experimentation in a generic way utilizing readily available ontologies in the web, not carefully constructed by hand ontologies. Our objective was to analyze the semantic disambiguation process, to see if it works satisfactorily, and which components can be improved with different algorithms.

To assess our relatedness measure's usefulness, 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. As Budanitsky and Hirst [1] found in a study comparing WordNet similarity measures human judgments give the best assessments of the "goodness" of a measure.

We have obtained relatedness judgments from 20 human subjects, 10 from the computer science field that had knowledge of the domain ontologies and 10 from the liberal arts field, that were used for the evaluation, for 25 pairs of concepts that we meet in 3 OWL domain ontologies freely available on the web, for the domains of **soccer**, **wine** and **people with pets**. 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". After calculating the mean ratings from the experiments on the concept pairs produced by the human ratings and the ratings the equations 3, 4, 10 and 11 produced for the three ontologies, we present the absolute values of the coefficients of correlation between the ratings in **Table 1**. We also present in this table, the overall satisfaction of the users after presenting them the results of the OntoNL Semantic Ranking procedure for the pairs of concepts used for the experimentation. The users were showed the semantically related concepts to the source-initial concept accompanied with the value of relatedness and the users could evaluate if the ranking was correct to their sense of the domain.

We have observed that the ratings from human subjects that come from the liberal arts field were closer to the ratings from the Properties sub-measure (rel_{PROP}) and the Related Senses sub-measure (rel_{RS}). On the contrary, the human subjects that were aware of the structures of the tested domain ontologies (from the Computer Science field) came closer to the ratings from the Conceptual Distance sub-measure (rel_{CD}) and to the ratings from the Properties sub-measure (rel_{PROP}). The impact of each of the sub-measures expressed by the factors w_1 , w_2 and w_3 of the eq. 11 can generally be tuned after experimentation of each specific application in a particular domain, that is expressed by an OWL domain ontology.

The values of the factors f_1 (for rel_{C1}), f_2 (for rel_{C2}) of equation 3, w_1 (for rel_{PROP}), w_2 (for rel_{RS}) and w_3 (for rel_{CD}) of equation 11 are shown in **Table 2**. The experimentation pointed to some first conclusions that are the basis for the relative weights value calculation algorithm extraction. The parameters that we take into account are described next but are not limited to them since the evaluation tests and the methodology for the relative weights values extraction are ongoing:

Table 1. 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 2

Measure	Humans LibArts Field			Humans CompSc Field			User Satisfaction over Ranking (%)		
	Soccer	Wine	PP	Soccer	Wine	PP	Soccer	Wine	PP
rel_{RS}	0,938	0,967	0,953	0,908	0,925	0,917	80%	85%	83%
rel_{CD}	0,935	0,947	0,927	0,929	0,961	0,982	82%	90%	89%
rel_{PROP}	0,964	0,948	0,954	0,945	0,943	0,963	87%	85%	89%
rel_{OntoNL}	0,978	0,981	0,969	0,968	0,972	0,987	92%	95%	93%

Table 2. The values of the relative weights f_1 and f_2 of eq. 3 and w_1 (for rel_{PROP}), w_2 (for rel_{RS}) and w_3 (for rel_{CD}) 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	0,5	0,5	0,7	0,1	0,2
Wine	0,8	0,2	0,25	0,2	0,55
P 'n' P	0,8	0,2	0,45	0,2	0,35

The language the ontology uses for its terminology. When ontologies are used directly from their source (web) a major factor of the rel_{RS} 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 rel_{PROP} 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.

To summarize the observations over the experimentations, the application of the semantic relatedness measure on a number of OWL ontologies produced some first conclusions. 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. 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. 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. Alternatively, algorithms for automatic mappings should also be investigated.

When the concepts of the ontology have a great number of properties that specialize them over other concepts, like in the ‘Soccer’ Ontology then if the parameter rel_{PROP} takes a great weight of influence in the overall OntoNL measure, the user satisfaction over the OntoNL Semantic Ranking procedure increases up to almost 10% of the average satisfaction using the three parameters of the measure individually.

Also, the conceptual distance is a measure that has a great influence if the ontology depth is big because this means that 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). This observation was applied in the evaluation of the measure over the ‘Soccer’ and the ‘People with Pets’ Ontologies and produced very good results.

7 Conclusions

We have presented the OntoNL ontology-driven semantic ranking methodology for ontology concepts 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 results during disambiguation. The disambiguation procedure is automatic and quite promising, since it is linguistically as complete as possible in an automatic environment [3] and it is enhanced with information based on the domain that the request refers to. It is easily reusable in many domains since the only restrictions are the used language (English) and OWL as the standard language for representing ontologies of a specific domain.

The OntoNL semantic ranking methodology depends on the OntoNL semantic relatedness measure for OWL domain ontologies. The measure is based on 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 motivation of this work came from the absence of a general, domain-independent Natural Language Interface Generator with good results in the Natural Language Disambiguation process. 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 a

semantic relatedness measure over OWL ontologies that is general, domain independent and covers the lack of a systematic way for calculating asymmetric semantic relatedness of concepts.

Overall, we state that the semantic relatedness measure that leads to 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.

References

1. Budanitsky, A., Hirst, G. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics* 32(1): 13-47
2. Fellbaum, C. editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA.
3. Karanastasi, A., Zwtos, A., Christodoulakis, S. 2006. User Interactions with Multimedia Repositories using Natural Language Interfaces - *OntoNL: an Architectural Framework and its Implementation*, in *Journal of Digital Information Management - JDIM*, Volume 4, Issue 4, December 2006
4. 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.
5. 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.
6. Lin, D. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*.
7. Rada, R., Bicknell, E. 1989. Ranking documents with a thesaurus. *JASIS*, 40(5):304–310, September.
8. 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.
9. Resnik, P. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal, Canada.
10. Sussna, M. 1993. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and Knowledge Management (CIKM-93)*, pages 67–74, Arlington, Virginia.
11. Wu, Z., Palmer, M. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico