# Evaluating the OntoNL Framework: a Natural Language Interface Generator for Knowledge Repositories

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

**Abstract**

One of the essential activities when providing a software system in general, is to evaluate the system based on qualitative and quantitative measures. We present in this paper the design and implementation of an evaluation framework for the OntoNL Framework, a natural language interface generator for knowledge repositories. We provide the definition and description of the measures, the methodology of evaluation and the results using an application of the OntoNL Framework. The evaluation framework has been based on the standard, ISO 9126, which is concerned primarily with the definition of quality characteristics to be used in the evaluation of software products.

## Categories and Subject Descriptors

H.3.4 [Systems and Software]: Performance Evaluation; I.2.1 [Applications and Expert Systems]: Natural Language Interfaces

## General Terms

Measurement, Performance, Experimentation, Human Factors

## Keywords

Natural Language Interface, Evaluation

# 1 Introduction

One of the essential activities when providing a software system in general, is to evaluate the system based on qualitative and quantitative measures. We present in this paper the design and implementation of an evaluation framework with methodologies for evaluating the efficiency and performance of the modules and methodologies of the OntoNL Framework, a natural language interface generator to knowledge repositories (Karanastasi, 2006). We present briefly the OntoNL Framework for building natural language interfaces to semantic repositories, as well as 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 of the application repositories.

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 the SPARQL 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, yes/no questions and WH-questions (who, were, what, etc.). The output for a particular natural language input 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.

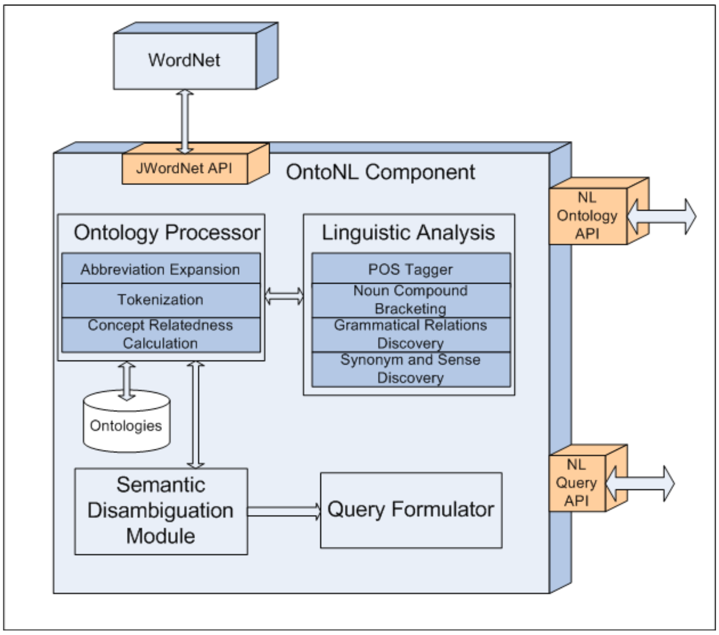

**Figure 1: The architecture of the OntoNL Framework**

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 also been used to help the evaluation methodology. The reference ontologies we used is an application of the DS-MIRF ontological infrastructure (Tsinaraki, 2004) and the WordNet for the syntactic analysis.

## 3 The Evaluation Framework

We have considered as a starting point an existing standard, ISO 9126 [http://www.issco.unige.ch/projects/ewg96/node13.html], which is concerned primarily with the definition of quality characteristics to be used in the evaluation of software products. ISO 9126 sets out six quality characteristics, which are intended to be exhaustive. From this it follows that each quality characteristics is very broad. Taking into account information from the ISO 9126 Standard we can summarize and broadly distinguish three measures of evaluation, appropriate to three different goals.

**Adequacy Evaluation:** This is determination of the fitness of a system for a purpose---will it do what is required, how well, at what cost, etc. Typically for a prospective user, it may be comparative or not, and may require considerable work to identify a user's needs.

**Diagnostic Evaluation:** This is production of a system performance profile with respect to some taxonimization of the space of possible inputs. It is typically used by system developers, but sometimes offered to end-users as well. It usually requires the construction of a large and hopefully representative test suite.

**Performance Evaluation:** This is measurement of system performance in one or more specific areas. It is typically used to compare like with like, whether two alternative implementations of a technology, or successive generations of the same implementation. It is typically created for system developers and/or R&D programme managers.

# 4 Measures Description

## 4.1 Adequacy Evaluation

The Adequacy Evaluation can be divided in two further evaluations: the Expert-based and the User-based evaluation. The Expert-based evaluation is performed by HCI experts who evaluate the usability of the interfaces according to a defined set of heuristics. These heuristics address mainly the Natural Language Interfaces usability. The user interface (UI) can be critical to the success or failure of a computer system. The development of UIs requires an iterative design and evaluation process involving users at every stage.

Specifically, the most significant parts to be considered are:

- developing a UI in a flexible, iterative manner, working in close collaboration with the users;
- identifying who will use the system, the tasks they want to carry out and the environment in which they will be working;
- creating a conceptual design;
- choosing the most appropriate interaction style;
- choosing appropriate interaction devices;
- using text, colour, images, moving images and sound effectively;
- evaluating the UI,

This particular type of evaluation concerns the graphical user interface of an application that makes use of the OntoNL Framework and it is not a subject of this paper. It would aim to measure the satisfaction of users for the effectiveness of applying their requests to a system using an information repository after presenting them the results.

## 4.2 Diagnostic Evaluation

The Diagnostic Evaluation is about testing the range of possible sentences that the OntoNL system can parse and disambiguate linguistically. It is conducted by system developers and it refers to the successfully parsing of natural language expressions and to different categories of grammatical relations combinations that need to be disambiguated. Below we present the different categories of request types that the system disambiguates and obtains results, through a class diagram and a description of the diagram (figure 2).

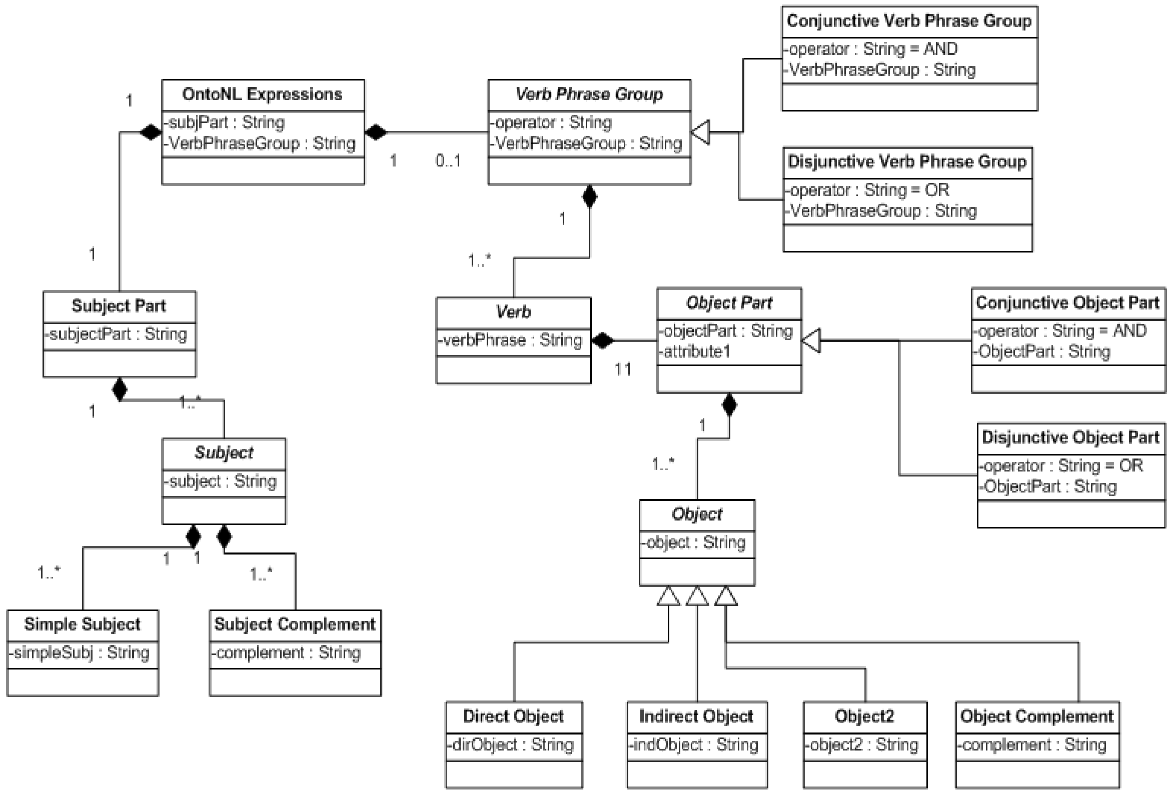

**Figure 2: The language model that describes the different categories of Natural Language expressions that the OntoNL can parse**

### 4.3 Performance Evaluation

The performance evaluation can be distinguished to the quantitative and the qualitative evaluation. We are interested in the qualitative performance.

The qualitative performance evaluation concerns the performance of the relatedness measure and the query formulation. Evaluation of semantic relatedness measures remains an open question [Agirre and Rigau, 1997, Resnik, 1995, Hirst and St-Onge, 1998]. In our survey of literature on the topic, we have come across three prevalent approaches: mathematical analysis, comparison with human judgement, and application-specific evaluation.

The first approach (see, e.g., [Wei, 1993, Lin, 1998]) consists in a (chiefly) theoretical examination of mathematical properties of a measure, such as whether it is actually a metric, whether it has singularities, whether its parameter-projections are smooth functions, etc. Such analyses, in our opinion, may certainly aid the comparison of several measures but perhaps not so much their individual assessment.

The second approach, comparison with human judgments of relatedness, does not appear to suffer from the same limitations; in fact, it arguably yields the most generic assessment of the 'goodness' of a measure; however, its major drawback lies in the difficulty of obtaining such judgements (i.e., designing a psycholinguistic experiment, validating its results, etc.). In his [1995] paper, Resnik presented a comparison of the ratings produced by his measure simR (and a couple of others) with those produced by human subjects on a set of 30 word pairs from an experiment by Miller and Charles [1991]. The fact that others [Jiang and Conrath, 1997, Lin, 1998] followed his lead and employed the same modestly sized dataset in their work appears to be a testament to the seriousness of the problem.

Because of these deficiencies, we, generally, have to take sides with the remaining group of researchers who have chosen to evaluate their measures in the framework of a particular NLP application.

However, since the trend has been established and since we have also found a use for the results in our application-specific evaluation, we decided to have the measures implemented as part of the application-specific evaluationn along with the evaluation of the measures based on human subjects ratings that have been demonstrated in (Karanastasi, 2007i, 2007ii).

# 5 Evaluation Results

## 5.1 Diagnostic Evaluation

We are interested in the successful parsing of sentences with the syntax shown in figure 2. Direct comparison between our system and other dependency parsers like Minipar [Lin, 1998] and the Link Parser [Sleator and Temperlay, 1993] is complicated by differences between the annotation schemes targeted by each system, presumably reflecting variations in theoretical and practical motivations. The systems do not always agree about which words should be counted as the dependents of a particular sentence. Even when the systems agree about whether two words are in a dependency relation, they may diverge about the type of the dependency. Each system assigns dependency types from a different set of grammatical relations and it is not straightforward to establish mappings between these sets. Also, the names used for relations vary considerably, and the distinctions between different relations may vary as well. Such differences make it difficult to directly compare the quality of the three systems. The most salient difference between the schemes is the level of granularity. Carroll's scheme contains 23 grammatical relations, MiniPar 59, Link 106 and ours 22.

To provide a qualitative comparison, we tagged, with the three taggers, fifteen sentences chosen from the Brown Corpus. The sentences we examined (table 1) agree with the language model we have developed for the OntoNL Framework. In what follows, we present in figures 3, 4 and 5 the dependency graphs that are produced after the parsing of the sentence "Bills on ports and immigration were submitted by Senator Brownback". We chose this sentence as an illustrative example because it is short but shows typical structures like prepositional phrases, coordination and noun compounding. The dependency graph is a tree, a singly rooted directed acyclic graph with no re-entrances. The graph representing Minipar output collapses directed paths through preposition nodes. It also adds antecedent links to 'clone' nodes between brackets. The graph for the Link Parser presents the same collapsing of directed paths through preposition nodes.

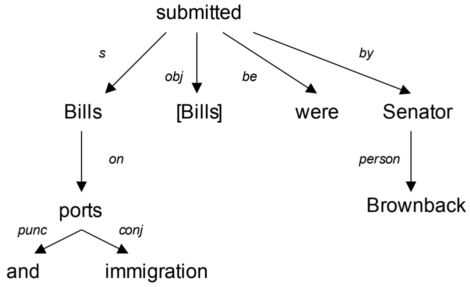

**Figure 3: Minipar's dependency parse for the sentence "Bills on ports and immigration were submitted by Senator Brownback"**

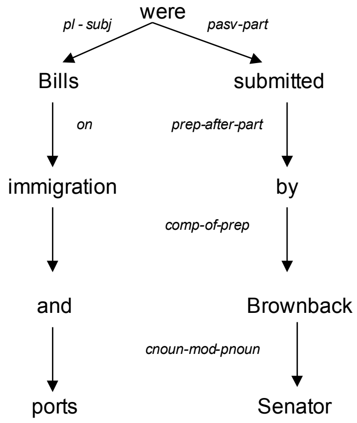

**Figure 4: Link Parser’s dependency parse for the sentence “Bills on ports and immigration were submitted by Senator Brownback”**

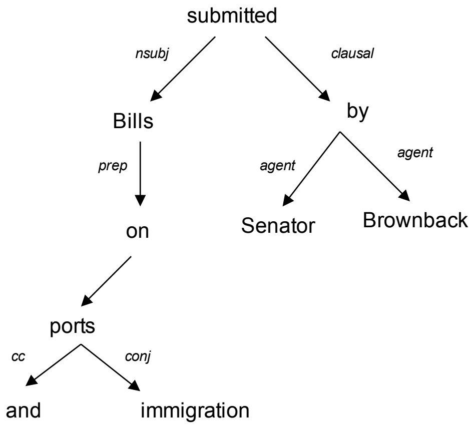

**Figure 5: OntoNL Parser’s dependency parse for the sentence “Bills on ports and immigration were submitted by Senator Brownback”**

Generally, the Stanford tagger (http://nlp.stanford.edu/software/tagger.shtml ), the tagger we used in our system and the Link tagger lead to more accurate structures than Minipar. The Stanford tagger was trained on the Penn Wall Street Journal Treebank and does a poor job at parsing questions, though. This is easily explained by the fact that the parser is trained on the Wall Street Journal section of the Penn Treebank in which not many questions occur. Minipar is confused by punctuation (already mentioned in [Lin, 1998]) and is also confused by conjunctions. Our parser behaves very well in conjunctions because of the strict language model it follows. An advantage of the Minipar is its capacity to identify collocations. The Link parser also has trouble with conjuction: it did not parse correctly sentences 6 and 15. We evaluated our system on this sample of 15 sentences. We obtained a dependency accuracy of about 80%. However it can be only considered as a rough estimate because of the quite small sample size and the complexity of the sentence structure. Our objective was to evaluate the OntoNL parsing mechanism in comparison with other well known parsers in order  to conclude to advantages and future refinements.

| | |
|---|---|
| 1 | She lived and was given a name.<br>ID: cm05 \| genre: scifi |
| 2 | He had better write a postcard to Walter.<br>ID: cn19 \| genre: adventure |
| 3 | People came in and out all evening to see the baby.<br>ID: cp02 \| genre: romance |
| 4 | Spencer said nothing.<br>ID: cp07 \| genre: romance |
| 5 | They make us conformists look good.<br>ID: cp15 \| genre: romance |
| 6 | A cookie with caramel filling and chocolate frosting won the cooking competition.<br>ID: ca30 \| genre: reportage |
| 7 | Everywhere I went in Formosa I asked the same question<br>ID: cb23 \| genre: editorial |
| 8 | The letters of the common soldiers are rich in humor.<br>ID: cf18 \| genre: popularlore |
| 9 | This time he was making no mistake<br>ID: cg32 \| genre: belles-lettres |
| 10 | It usually turned out well for him<br>ID: cg60 \| genre: belles-lettres |
| 11 | The author of the anonymous notes seemed to be all-knowing.<br>ID: cn11 \| genre: adventure |
| 12 | Below he could see the bright torches lighting the riverbank |

| | |
|---|---|
| | ID: ck21 \| genre: generalfiction |
| 13 | Beckworth handed the pass to the colonel.<br>ID: ck21 \| genre: generalfiction |
| 14 | Must Berlin remain divided?<br>ID: cb02 \| genre: editorial |
| 15 | Old, tired, trembling the woman came to the cannery.<br>ID: cb08 \| genre: editorial |

**Table 1: 15 sentences from the Brown Corpus, to compare outputs of Minipar, the Link Parser and the OntoNL parser.**

## 5.2 Performance Evaluation

The Performance Evaluation is comprised of the two parts of evaluation; the quantitative and the qualitative evaluation. In this paper we are going to deal with the qualitative evaluation of specific processes of the OntoNL Framework. The qualitative evaluation concerns measuring the effectiveness of the **noun compound bracketing mechanism**, the **semantic relatedness measurement** (Karanastasi, 2007i, 2007ii) and an **application-based evaluation** of measures of relatedness. In this paper we describe the noun compound bracketing mechanism and the application based evaluation.

### 5.2.1 Noun Compound Bracketing

In this section we wil define the methodology of training the noun compound bracketing algorithm and evaluating its accuracy by using two large OWL domain ontologies freely available in the web, the Soccer Ontology (http://www.music.tuc.gr/ontologies/mpeg7/mds/socccer/) and the Biopax-Level 2 Ontology (http://www.biopax.org/ ).

In all the experimental work we will only consider English compound nouns. Nonetheless, compounds appear in many other languages and there seems no reason why the same techniques we used would work less well in these. We also assume that the possible compound has been recognised from the surrounding text based on the linguistic, so that the system is presented with a sequence of nouns known to be a compound.

**Method:** Given an identified compound, it is simplest to define the parsing task as one of bracketing. That is, the system must select the most likely binary bracketing of the noun sequence, assuming that it is a compound noun.

According to most views of compounding, the composition of two or more nouns yields an element with essentially the same syntactic behaviour as the original nouns. An n-word compound noun acts exactly like a single noun, as do three word compounds and so forth.

To define the primary goal of the work in the OntoNL noun compound bracketing mechanism we conclude to the next statement:

**Problem Statement**: Given a three word English compound noun predict whether the most likely syntactic analysis is left-branching or right-branching.

**Extracting a Test Set:** Two test sets of syntactically unambiguous noun compounds was extracted from a 67 pages document describing molecular binding interactions, protein post-translational modifications, basic experimental descriptions, and hierarchical pathways and a 115 official document from FIFA describing the rules of football in the following way. Because the corpus is not tagged or parsed, a somewhat conservative strategy of looking for unambiguous sequences of nouns was used. To distinguish nouns from other words we used once again the Stanford Log-Linear Tagger to generate the set of words that can only be used as nouns. Let’s call this set from now on N. All consecutive sequences of these words were extracted, and the three word sequences used to form the test set. The result was 98 test trigrams.

These triples were manually analysed using as context the entire article in which they appeared. In some cases, the sequence was not a noun compound (nouns can appear adjacent to one another across various constituent boundaries) and was marked as an error. Other compounds exhibited

*SEMANTIC INDETERMINACY* where the two possible bracketings cannot be distinguished in the context. The remaining compounds were assigned either a left-branching or right-branching analysis. The number of each kind is shown in 2.

| Type | Number | Proportion |
|---|---|---|
| Error | 7 | 7% |
| Indeterminate | 11 | 11% |
| Left-branching | 52 | 53% |
| Right-branching | 28 | 29% |

**Table 2: Test Set distribution**

**Conceptual Association:** We use the term Conceptual Association in this study to refer to association values computed between groups of words. We have used groups consisting of all categories from the Roget's II: The New Thesaurus (http://www.bartleby.com/62/). By assuming that all words within a group behave similarly, the parameter space can be built in terms of the groups rather than in terms of the words.

Given two thesaurus categories t1 and t2, there is a parameter which represents the degree of acceptability of the structure [n1 n2] where n1 is a noun appearing in t1 and n2 appears in t2. By the assumption that words within a group behave similarly, this is constant given the two categories.

Following Lauer (1995) we can formally write this parameter as Pr(t1→ t2) where the event t1→ t2 denotes the modification of a noun in t2 by a noun in t1.

**Training:** To ensure that the test set is disjoint from the training data, all occurrences of the test noun compounds have been removed from the training corpus.

We are going to explore two types of training scheme. The first employs a pattern that follows Pustejovsky (1993) in counting the occurrences of subcomponents. A training instance is any sequence of four words w1w2w3w4 where $w_1, w_4 \notin N$ (N is a set of words that can be used only as nouns) and $w_2, w_3 \in N$. Let $count_p(w_1, w_2)$ be the number of times a sequence w1w2w3w4 occurs in the training corpus with $w_1, w_4 \notin N$.

The second type uses a window to collect training instances by observing how often a pair of nouns co-occur within some fixed number of words. In this work, a variety of window sizes are used.

In OntoNL we used a window to collect training instances by observing how often a pair of nouns co-occurs within some fixed number of words. For window size $n \geq 2$, let $count_n(w_1, w_2)$ be the number of times a sequence n1w1…win2 occurs in the training corpus where $i \leq n-2$. The estimates are:

$$\mathrm{P}(t_1 \rightarrow t_2) = \frac{1}{\sum_{w_1 \in N, w_2 \in t_2} \frac{count_n(w_1, w_2)}{amb(w_1, w_2)}} \sum_{w_1 \in t_1, w_2 \in t_2} \frac{count_n(w_1, w_2)}{amb(w_1) amb(w_2)}$$

where amb(w) counts the number of categories w appears and N is a set of words that can only be used as nouns. The amb(w) has the effect of dividing the evidence from a training instance across all possible categories for the words. The first parameter of the multiplication is used to ensure that the parameters for a head noun sum to unity. After the calculation of the estimates, we continue by trying to make a right choice of all possible analyses for three word compounds, which are the counting of a right or a left branching analysis. So, for the adjacency model and a given compound of w1, w2, w3 the estimation of the ratio is done by applying the equation

$$R_{adj} = \frac{\sum_{t_i \in cats(w_i)} P(t_1 \rightarrow t_2)}{\sum_{t_i \in cats(w_i)} P\left(t_2 \rightarrow t_3\right)}$$

for the dependency model and a given compound of w1, w2, w3 the estimation of the ratio is done by applying the equation

$$R_{dep} = \frac{\sum_{t_i \in cats(w_i)} P(t_1 \to t_2) P(t_2 \to t_3)}{\sum_{t_i \in cats(w_i)} P(t_1 \to t_3) P(t_2 \to t_3)}$$

where t1, t2 and t3 are conceptual categories in a taxonomy or thesaurus, and the nouns w1, …,wn are members of these categories. If the ratio is >1 then we conclude to a left-branching analysis. If the ratio is <1 then a right branching analysis is chosen. If it is =1, the OntoNL analyzer, based on Lauer (Lauer, 1995) guesses left-branching, a rare case for conceptual association based on experimental results.

For a correct result we must sum over all possible categories for the words in the compound. In any case, the estimation of probabilities over concepts reduces the number of model parameters.

**Results:** In what follows, all evidence used to estimate the parameters of the model is collected in one pass over the corpus and stored in a fast access data structure. Evidence is gathered across the entire vocabulary, not just for those words necessary for analysing a particular test set. Once trained in this way, the program can quickly analyse any compound, restricted only by the lexicon and thesaurus. This demonstrates that the parsing strategy can be directly employed using currently available hardware in broad coverage natural language processing systems.

Six different training schemes have been used to estimate the parameters and each set of estimates used to analyse the test set under both the adjacency and the dependency model. The schemes used are the pattern that follows Pustejovsky (1993) in counting the occurrences of subcomponents and windowed training schemes with window widths of 2, 3, 4, 5 and 10 words.

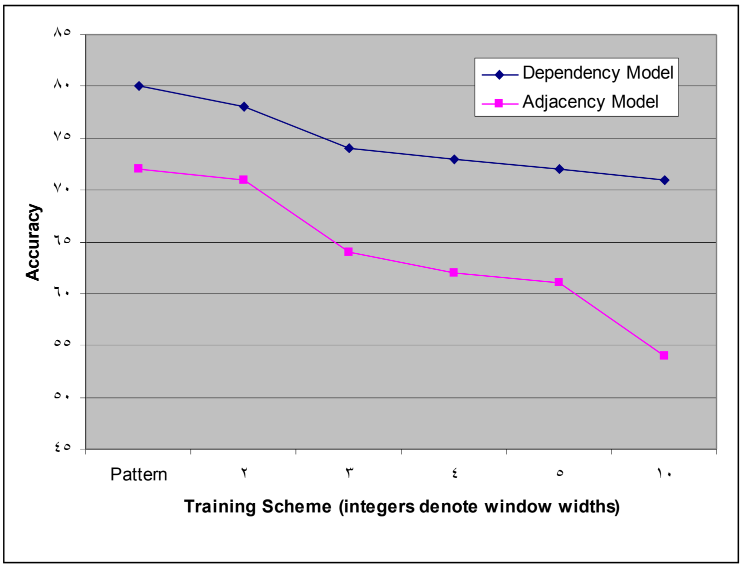

**Figure 6: Accuracy of analysis of the test set under the dependency and the adjacency model for the pattern training scheme that follows Pustejovsky (1993) in counting the occurrences of subcomponents and for the windowed training schemes with window widths of 2, 3, 4, 5 and 10 words**

The accuracy on the test set for all these experiments is shown in Figure 6. As can been seen, the OntoNL dependency model is more accurate than the OntoNL adjacency model. The proportion of cases in which the procedure was forced to guess, either because no data supported either analysis, is quite low. For the pattern and two-word window training schemes, the guess rate is less that 6% for both models. In the three-word window training scheme, the guess rate is less that 2%. For all larger windows, neither model is ever forced to guess.

In no case do any of the windowed training schemes outperform the pattern scheme. It seems that additional instances admitted by the windowed schemes are too noisy to make an improvement.

**Lexical Association:** To determine the difference made by conceptual association, the pattern training scheme has been retrained using lexical counts for both the dependency and adjacency model, but only for the words in the test set. Accuracy and guess rates are shown in figure 4. Conceptual association outperforms lexical association, presumably because of its ability to generalize (see Figure7).

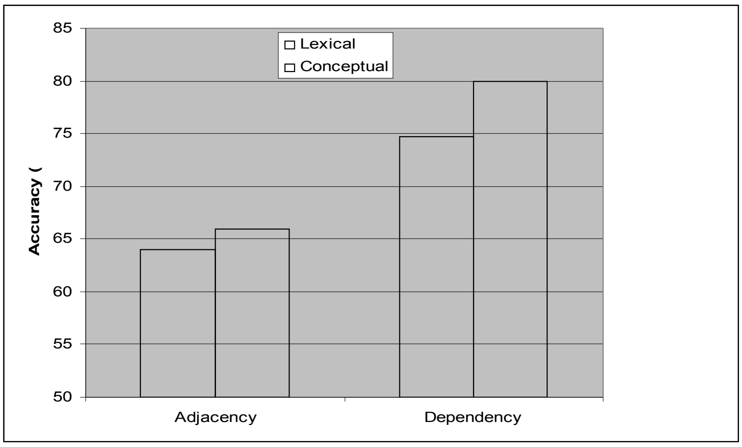

**Figure 7: Accuracy of analysis of the test set under the dependency and the adjacency model for the pattern training scheme using lexical association and conceptual association**

**Using a Tagger:** One problem with the training methods we presented previously is the restriction of training data to nouns in N. Many nouns, especially common ones, have verbal or adjectival usages that preclude them from being in N. Yet when they occur as nouns, they still provide useful training information that the current system ignores. To test whether using tagged data would make a difference, the freely available Stanford Log-Linear POS Tagger was applied to the corpus. Since no manually tagged training data is available for our corpus, the tagger's default rules were used.

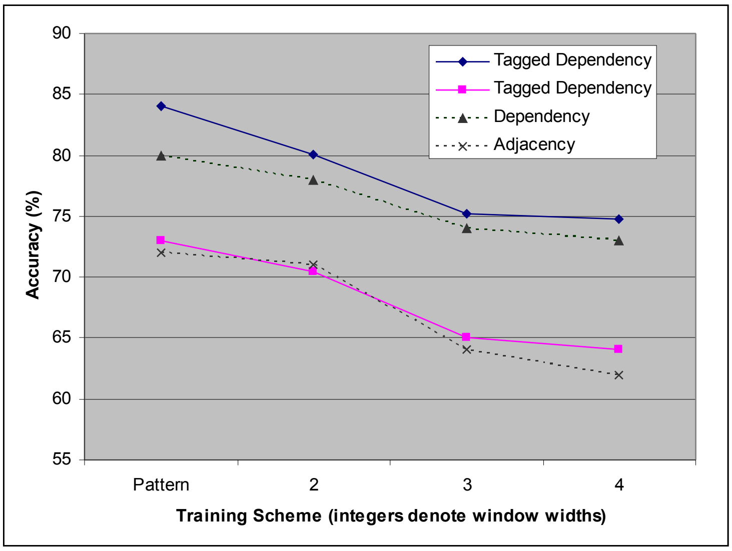

**Figure 8: Accuracy of analyzing the test set using a tagged corpus under the dependency and the adjacency model for the pattern training scheme that follows Pustejovsky (1993) in counting the occurrences of subcomponents and for the windowed training schemes with window widths of 2, 3, and 4 words and comparison with the accuracy presented in figure 6.**

Four training schemes have been used and the tuned analysis procedures applied to the test set. Figure 8 shows the resulting accuracy, with accuracy values from figure 6 displayed with dotted lines. If anything, admitting additional training data based on the tagger introduces more noise, reducing the accuracy. However, for the pattern training scheme an improvement was made to the dependency model, producing the highest overall accuracy of 84%.

**Using Domain Ontologies:** What we propose in this method is to use as corpus the nouns used for naming the concepts of the domain ontolog, plus their synonyms and the descriptions of the concepts inside the ontology. One problem with this approach is that in the absence of descriptions we only have as training corpus the names of the concepts of the ontology which is a very limited corpus with either excellent or very bad results. On the other hand when the domain ontology used for the semantic disambiguation is also used for the noun compound bracketing mechanism there is no need to find relative to the domain corpuses each time we want to use the OntoNL.

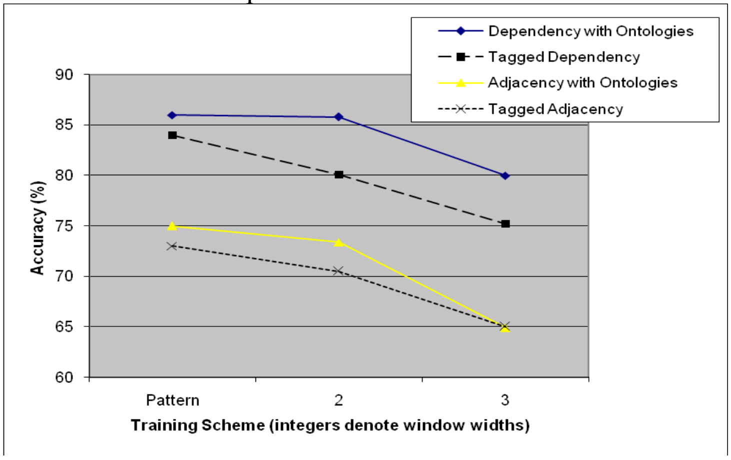

**Figure 9: Accuracy of analyzing the test set using a tagged corpus and domain ontologies under the dependency and the adjacency model for the pattern training scheme that follows Pustejovsky (1993) in counting the occurrences of subcomponents and for the windowed training schemes with window widths of 2 and 3 words and comparison with the accuracy presented in figure 8.**

Three training schemes have been used and the tuned analysis procedures applied to the test set. Figure9 shows the resulting accuracy, with accuracy values from Figure8 displayed with dotted lines. What we see is that the resulting accuracy is better in all cases and that the most significant improvement was in the dependency model with training scheme of window width 2 (85,8% from 80,1%).

5.2.2 An application-based evaluation of measures of relatedness

We have performed an application-based evaluation of the OntoNL Semantic Relatedness Measure. The application used the OWL Ontology for the domain of soccer (http://lamia.ced.tuc.gr/ontologies/AV_MDS03/soccer ), because it is a large and very specific ontology. Also, the context of the ontology is familiar with the users.

We first asked the users to submit requests. We gathered a total of 60 requests, after eliminating any duplicates. We distinguished the types of expressions based on the OntoNL Language Model in 3 different types:

1. Subject Part
2. Subject Part – Conjuctive/Disjunctive/Plain Verb Phrase
3. Subject Part – Verb – Conjuctive/Disjunctive/Plain Object Part

We have presented to the human subjects, the resulted concepts related to the subject concept of their request. The users replied the ranking position of their correct response in mind and this experiment was conducted twice. Since our results are a ranked list, we use a scoring metric based on the inverse rank of our results, similar to the idea of Mean and Total Reciprocal Rank scores described in [Radev et al, 2002], which are used widely in evaluation for information retrieval systems with ranked results. Hence our precision and recall are defined as:

$$PRECISION = \frac{\square \frac{1}{ranking}}{\# requests}$$

$$RECALL = \frac{n(accepted\_ranking)}{\# requests}$$

The precision is depended on the ranking position of the correct related concept to the subject concept of the request. The recall is depended on the number of the related concepts the algorithm returns. In Table 3 we present the precision and recall scores we obtained for the two most complex datasets of request types.

| **DataSet** | **Precision** | **Recall (n = 3)** | **Recall (n =5)** | **Recall (n =8)** |
|---|---|---|---|---|
| Subject Part –<br>Conjuctive/Disjunctive/Plain Verb Phrase (15 requests) | 49% | 60% | 86,7% | 100% |
| Subject Part –<br>Verb –<br>Conjuctive/Disjunctive/Plain Object Part (25 requests) | 39,7% | 52% | 76% | 92% |
| Total | 44% | 55% | 80% | 95% |

**Table 3: Quality metrics for the first iteration**

What we see is that overall we gain more than 50% of the correct matches in the first three hits and that the requests of type Subject Part – Conjuctive/Disjunctive/Plain Verb Phrase had better precision and recall than the requests of type Subject Part – Verb – Conjuctive/Disjunctive/Plain Object Part requests. This is because we use the verbs in this application to disambiguate in a more sufficient way the RelationTypes modeled in the OWL Domain Ontology for Soccer that is based on the MPEG-7 (Tsinaraki, 2004).

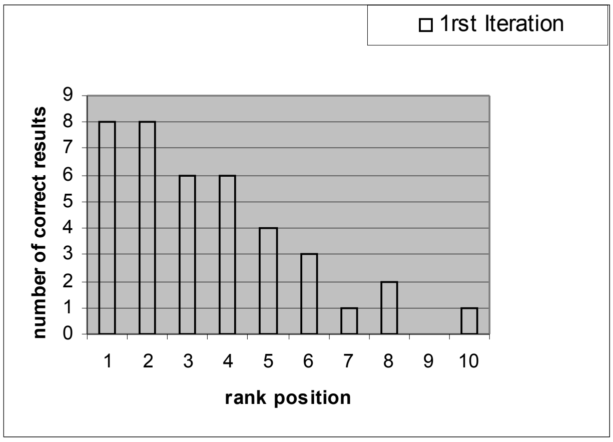

**Figure 10: The precision of the OntoNL measure to the user input for the requests**

After this experiment we asked the users to submit new requests and we once again gathered 20 requests. In Table 4 we present the precision and recall scores we obtained for the two most complex datasets of request types and for a second iteration of the experiment.

| **DataSet** | **Precision** | **Recall (n = 3)** | **Recall (n =5)** | **Recall (n =7)** |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Subject Part –<br>Conjuctive/Disjunctive/Plain Verb Phrase (10 requests) | 47% | 70% | 90% | 100% |
| Subject Part –<br>Verb –<br>Conjuctive/Disjunctive/Plain Object Part (10 requests) | 46,1% | 60% | 100% | - |
| Total | 46,63% | 65% | 90% | 100% |

**Table 4: Quality metrics for the second iteration**

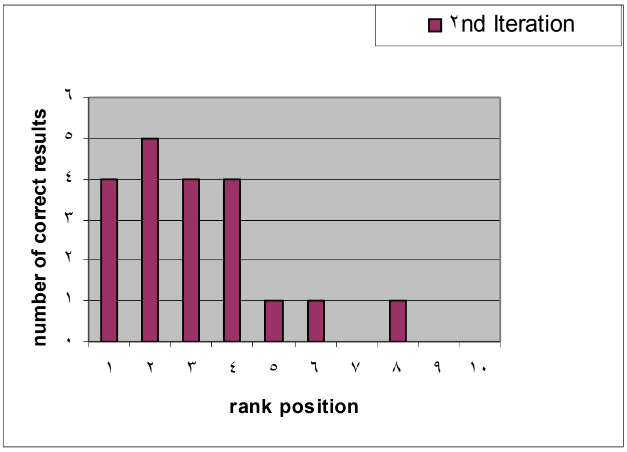

**Figure 11: The precision of the OntoNL measure to the user input for the requests of disambiguation type (2) for a second iteration**

What we see here is that in total we gain a 65% of the correct matches in the first 3 results of the OntoNL Disambiguation Procedure an a 90% in the first 5 results. The overall conclusion that derives is that in a second iteration of tests the performance was better because of the familiarity of the users using the system increased. In more details, the request type Subject Part – Conjuctive/Disjunctive/Plain Verb Phrase has a better precision but the request type Subject Part – Verb – Conjuctive/Disjunctive/Plain Object Part has a better recall.

We also present the overall satisfaction of users with respect to the effectiveness of the results compared against a keyword-based search (Figure 12). Overall, the performance decreases a little as the complexity of the language model increases, but as shown in Figure 12, we get the correct results sooner and faster against a keyword-based search.

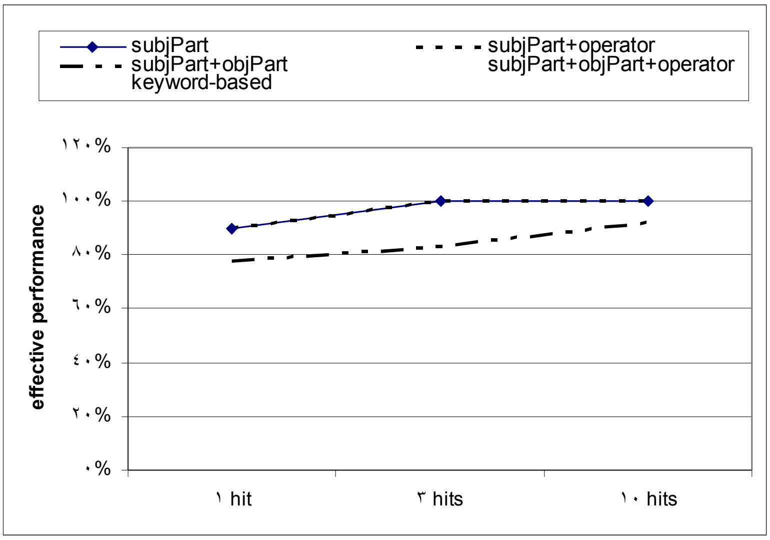

**Figure 12: The effectiveness of the NL2DL in the domain of soccer against a keyword-based search**

## 4 Conclusions

In this paper we have presented the evaluation framework that has been developed for measuring the efficiency of different parts of the OntoNL Framework, a Natural Language Interface Generator for Knowledge Repositories. The evaluation framework is based on information from the ISO 9126 Standard and we summarized three measures of evaluation, appropriate to three different goals.

The **Adequacy Evaluation**, that stands for the determination of the fitness of a system for a purpose---will it do what is required, how well, at what cost, etc. Typically for a prospective user, it may be comparative or not, and may require considerable work to identify a user's needs.

The **Diagnostic Evaluation**, that measures the system performance profile with respect to some taxonimization of the space of possible inputs. It is typically used by system developers, but sometimes offered to end-users as well. It usually requires the construction of a large and hopefully representative test suite.

The **Performance Evaluation**, that measures the system performance in one or more specific areas. It is typically used to compare like with like, whether two alternative implementations of a technology, or successive generations of the same implementation. It is typically created for system developers and/or R&D programme managers.

We examined how this framework fit the needs of the OntoNL Framework by presenting the definition of the procedures and algorithms that we evaluated and by showing the results after the evaluation tests.

## 5 References

Agirre, E., Rigau, G. 1997. A proposal for word sense disambiguation using conceptual distance. In Recent Advances in Natural Language Processing: Selected Papers from RANLP'95, volume 136 of Amsterdam Sudies in the Theory and History of Linguistic Science: Current Issues in Linguistic Theory, chapter 2, pages 161-173. John Benjamins Publishing Company, Amsterdam/Phildadelphia, 1997.

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.

Hirst, G., St-Onge, D. 1998 Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, WordNet: An Electronic Lexical Database, chapter 13, pages 305-332. The MIT Press, Cambridge, MA, 1998.

Karanastasi, A., Christodoulakis, S. 2007 The OntoNL Semantic Relatedness Measure for OWL Ontologies, in the Proceedings of the Second IEEE International Conference on Digital Information Management (IEEE ICDIM), 28-31 October 2007, Lyon, France

Karanastasi, A., Christodoulakis, S. 2007 Semantic Processing of Natural Language Queries in the OntoNL Framework, in the Proceedings of the IEEE International Conference on Semantic Computing (IEEE ICSC), 17-19 September

Karanastasi, 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

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

Lin, D. 1998. An Information-Theoretic Definition of Similarity. In Proceedings of the 15th International Conference on Machine Learning, Madison, WI

Miller, G. A., Charles, W. G. 1991. Contextual Correlates of Semantic Similarity. Language and Cognitive Processes 6, 1-28.

Radev, D., Qi, H., Wu, H., Fan, W. 2002 Evaluating Web-based Question Ansering Systems. Proceedings of LREC, 2002

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of the l4th International Joint Conference on Artificial Intelligence (1JCAI- 95), pages 448-453.

Sleator, D., Temperlay, D. 1993. Parsing English with a link grammar. In Third International Workshop on Parsing Technologies.

Tsinaraki C., Polydoros P., Christodoulakis S., 2004 Interoperability support for Ontology-based Video Retrieval Applications, in the Proceedings of CIVR 2004, Dublin/Ireland, July 2004

Wei, M. 1993 An analysis of word relatedness correlation measures. Master's thesis, University of Western Ontario, London, Ontario, May 1993.