

User Interactions with Multimedia Repositories using Natural Language Interfaces: an Architectural Framework and its Implementation

Anastasia Karanastasi, Alexandros Zotos, Stavros Christodoulakis

Lab. Of Distributed Multimedia Information Systems / Technical University of Crete (MUSIC/TUC)
University Campus, Kounoupidiana, Chania, Greece
[allegra, azotos,stavros]@ced.tuc.gr

We present a generalized architectural framework for constructing and using natural language interfaces for interactions with multimedia repositories. The framework can take advantage of user profiles to rank the resulting queries from the natural language expressions. This to avoid or reduce the disambiguation overhead, meaning the elimination of the possible senses that can be assigned to a word in the discourse in a particular domain. The architecture for the disambiguation procedure is based on a combination of classical search techniques and a constrained spreading activation algorithm, an algorithm for searching information in networks like ontologies. We calculate the strength of relations between concepts in ontologies by assigning a weight value. The constrained spreading activation algorithm used is controlled by heuristic rules that are obtained from the search mechanism applied in the ontology. We describe the implementation of this framework for supporting interactions with a multimedia repository, described with the MPEG-7 MDS (Multimedia Description Schemes) structures.

1. Introduction

The advantages of natural language interfaces are well understood. They can be seen in user interaction with information repositories or when using modern interaction devices, like mobile phones, PDA's etc. and in general when the information is complex.

The extended use of the web has created a need for services that will help naive users to find information they need fast and without cost. One example is SCORE [Sheth et al., 2002], an interesting semantic searcher. It uses automatic classification and information-extraction techniques together with metadata and ontology information to enable contextual multi-domain searches. These searches try to understand the exact user information need expressed in a keyword query.

Another example is ONTOSEEK. ONTOSEEK is an information retrieval system coupled with an ontology [Guarino, 1999]. ONTOSEEK performs retrieval based on content instead of string-based retrieval. Queries are translated to conceptual graphs, but the problem in this step is according to the authors "in reducing to ontology-driven graph matching where individual nodes and arcs match if the ontology indicates that a subsumption relation holds between them". These graphs are semi-automatically constructed and users have to verify the links between different nodes in the graph via the designated user interface.

Another question-answering system which amalgamates Natural Language Processing (NLP), Logic, Ontologies and Information Retrieval techniques to provide answers to queries in a specific domain in real time is AQUA [Vargas-Vera and Motta, 2004]. AQUA translates English questions into logical queries that are then used to generate of proofs. AQUA is coupled with the AKT reference ontology for the academic domain. This ontology (written in OCML) currently contains people, organizations, research areas, projects, publications, technologies and events, and works as a pattern-matching, which means that it tries to find exact match with names in the ontology.

Until recently, the natural language interfaces (NLI) between humans and machines were either specific to a particular application with limited expectations or linguistic-based with possibly many ambiguities that led to lengthy disambiguation dialogues. An attempt of using a more generalized approach of the construction of an NLI, particularly in the domain of digital TV was presented by Karanastasi et. al [2003, 2004] with good results when dealing with ambiguities. There was no need of using clarification dialogues for the disambiguation, because of the well-structured domain of Digital TV, the TV-Anytime standard [<http://www.tv-anytime.org>] and the TV-Anytime User Profile information. A limitation of such a system is that it is not reusable. This limitation stems from using a domain grammar, with specific domain grammar rules. Also, the grammar rules are defined by the syntax of the repository they refer to, a fact that can be limiting in searching in more than one ontology that describes the same domain.

The proposed architecture uses OWL Web Ontology Language and word ontologies for the disambiguation of the user's query with a preprocessing phase for the linguistically representation of the ontology for better matching. The disambiguation is the assignment of the correct sense a word can take in a particular domain. The language model is as complete as possible from the linguistic part (syntactic and semantic based on a word ontology). The approach is to be able to retrieve concept instances (OWL individuals) that are strongly related to a word from user's request even if it is not appeared inside the concept. Also, we use User Profile information for better clustering based on the context of the ontologies and better ranking of the result queries.

2. The OntoNL System

The goal of the OntoNL system is to address the knowledge engineering bottleneck for natural language processing systems. To this end, we present the OntoNL natural language interface generator for interactions with multimedia repositories and make the following claims:

1. 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 consults user profiles to personalize the disambiguation
2. The OntoNL framework makes use of OWL rich vocabulary by using upper and domain ontologies. The design proposed here is especially useful in applications where the user searches for concept instances of the model and not for "arbitrary" data. That is, usually the keywords in the query denote one or more concepts. Given an OWL ontology, weights are assigned to links based on certain properties of the ontology, so that they measure the strength of the relation. In this way we can identify related concepts in the ontology to the ones retrieved by the user's request.
3. The OntoNL framework uses User Profiles to guide the semantic search in the domain ontology and to rank the results in a way a user meets his preferences.

To demonstrate support for these claims, we use OntoNL framework to create a Natural Language Interface for the domain of soccer that is used in a question answering system. The work presented in this paper has been carried out in the scope of the EU-funded Network of Excellence (IST-507618) DELOS II (www.delos.info).

2.1. The Framework Architecture

In this paper we describe a software engineering framework that aims to automate as much as possible the construction of natural language interfaces to multimedia repositories. We are targeting to the creation of a framework, since it provides a very high degree of reuse and it is easily extendable, which is significant in the automatic construction of natural language interfaces domain independent. In this way we face the problem of the prohibited cost of constructing natural language interfaces for particular applications and domain. The framework is an extendable subsystem for a set of related services. It is a cohesive set of abstract classes that define a natural language interface to conform to and object interactions to participate in. We have made a number of assumptions for the design of the framework, as they are listed below:

- Knowledge is stored in large repositories using a knowledge representation language such as OWL. The knowledge representation language is used to describe higher level knowledge concerning a large class of applications (such as knowledge described in standards), as well as domain specific knowledge (such as domain ontologies). For efficiency reasons the individuals (instances) of this knowledge structure may be stored in a different data management system such as an XML Database
- The users of such a system may have preferences expressed in profiles, and that the user profiles are also described with generic knowledge structures expressed in the same knowledge representation language (for example user profile structures are defined in several current standards like TV-Anytime and MPEG-7 [Salembier, 2001]). The individual user profiles are stored in the same data management system as the knowledge instances.

The basic structure of the framework as shown in figure 1, consists of a component responsible for the linguistic analysis of a user expression in English, a component for the semantic disambiguation based on the application's domain, a component for the processing of the ontologies that comprise the domain, a component for the reformulation of the disambiguated user expression to a knowledge manipulation language and a component for the result processing.

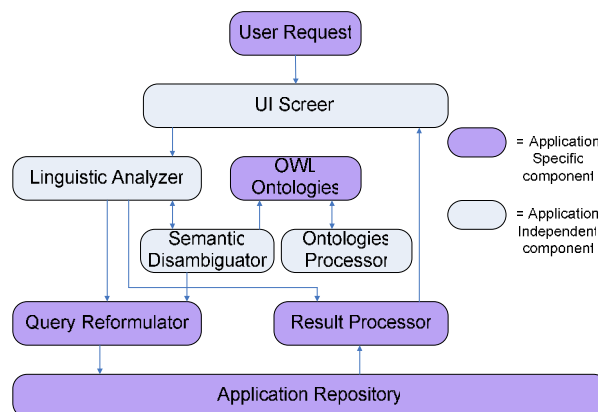


Figure 1. Framework Architecture

In the following subsections we are going to give details about the three most interesting components for the disambiguation procedure, the linguistic analyzer, the ontologies processor and the semantic disambiguator.

2.2. Linguistic Analyzer

Since the user input is natural language expressions, the natural language interface must be able to recognize English sentence elements and structure.

Specifically, when dealing with question-answering systems the sentences that a user uses are requests. Requests do not contain the actual information to address the knowledge repository in the subject of the sentence, but in one or more dependent clauses that complement the independent clause to a complex sentence. For example, in a domain concerning football, a user query could be:

- I want you to find me the players that scored for Barcelona in the last two football games that used to play for Milan.

In this case, the actual 'subject' of the request is not the one in the independent clause 'I want you to find me the players...', but the object of the independent clause is actually the 'subject' of interest of the user's request ('players'). So, we need to identify what the user asks the system and the additional preferences that he gives for this 'subject'. In the next sub-sections we describe the mechanisms that the parser uses to disambiguate syntactically and semantically English language interactions with the multimedia repositories.

Part-Of-Speech Tagging. The task of POS-tagging is to assign part of speech tags to words reflecting their syntactic category. Often words can belong to different syntactic categories in different contexts. Essentially then POS-tagging is a first attempt to disambiguate the sense of every word that constitutes the user's request. 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. In our system we adopted a maximum entropy approach, because it allows the inclusion of diverse sources of information without causing fragmentation and without necessarily assuming independence between the predictors [Toutanova et. al, 2003]. The part-of-speech tagger demonstrates the following ideas: (i) explicit use of both preceding and following tag contexts (past and future tag identity) via a dependency network representation, (ii) broad use of lexical features, including jointly conditioning on multiple consecutive words, (iii) effective use of priors in conditional log-linear models, and (iv) finegrained modeling of unknown word features. By implementing these ideas, the resulting Stanford tagger, implemented in Stanford University, gives 96.86% accuracy on the Penn Treebank, an error reduction of 4.4% on the best previous single automatically learned tagging result and 86.91% on previously unseen words.

Noun Compound Analysis. The methodology we use when dealing with noun compounds is that we, first, use a method to expand n-grams into all morphological forms by the use of morphological tools [Minnen, et. al., 2001]. For example, if we have a bigram 'player scores', then we create a list of all possible forms: 'player scores', 'player score', 'players score', etc. Between the two most significant models for syntactically analyzing noun compounds we use the dependency model [Lauer, 1995] over the adjacency [Marcus, 1980], based on its successful performance in previous applications. The training corpus is the set of the ontologies that describe each different domain. This may lead to the conclusion that the test set is very limited in comparison to a linguistic corpus, but it is more accurate to a specific domain. By combining the use of domain ontologies as the training corpus and the WordNet by taking advantage of the hyponyms and synonyms, we maintain the needed information. We use hyponyms as groups of words with similar behavior, so to limit the parameter space in terms of the groups and limit the vast amount of memory space needed for applying lexical association to noun compounds [Resnik, 1993]. The test set in this case is comprised by the noun compounds that may appear in the user input. The procedure of the noun compound bracketing is useful in determining correctly the grammatical relationships that structure the language model.

Sentence Patterns: Locating Grammatical Relationships. Grammatical relationships are an important aspect of natural language processing. Relationships, such as subject, object, conjunction, etc. are the semantic basis for the information extraction role of the system. These relationships also help in conducting user-friendly answers with the retrieved data to address the user. The extraction of these relationships is not a result of a training procedure, but a procedure of modeling grammar rules of English inside the Natural Language Parser. It is not the purpose of this work to eliminate all the possible combinations of the Subject-Verb-Object theory, but help the system for the better information extraction.

The annotation scheme for syntactic information, is based on grammatical relations that are composed of bilocal dependencies (between a head and a dependent) labelled with the name of the relation involving the two words.

- **arg(head, dependent)** - The most generic relation between a head and an argument
- **subj/dobj(head, dependent)** - A specialization of the relation arg which can instantiate either subjects or direct objects.
- **subj(head, dependent, initial_gr)** - The relation between a predicate and its subject; where appropriate, the initial_gr indicates the syntactic link between the predicate and subject before any GR-changing process.
- **comp(head, dependent)** - The most generic relation between a head and complement.
- **obj(head, dependent)** - The most generic relation between a head and object.
- **dobj(head, dependent, initial_gr)** - The relation between a predicate and its direct object – the first non-clausal complement following the predicate which is not introduced by a preposition; initial_gr is iobj after dative shift.
- **iobj(type, head, dependent)** - The relation between a predicate and a non-clausal complement introduced by a preposition; type indicates the preposition introducing the dependent.
- **obj2(head, dependent)** - The relation between a predicate and the second non-clausal complement in ditransitive constructions.
- **clausal (head, dependent)** - The most generic relation between a head and a clausal complement.
- **nscomp (type, head, dependent)** - The relation between a predicate and a clausal complement which has no overt subject. The type slot indicates the complementiser/preposition, if any, introducing the XP.
- **scomp (type, head, dependent)** - The relation between a predicate and a clausal complement which has an overt subject.

We have developed a rule-based parser and a handcrafted domain-specific unification grammar. The output of this rule-based system is a syntactic feature structure corresponding to the input sentence. Extracting grammatical relations from the feature structure produced by the parser is simple: there is a grammatical relation between the head word of each sub-structure and the head word of the outer structure containing the sub-structure in question. Each grammatical relation is named after the syntactic function of the sub-structure in relation to its outer structure. The lexicon in this case is a file automatically constructed by the utterance the user inserts in our system. The grammar includes grammar rules that act like patterns to the structure the input utterance has. Then we look for a match between the structure of the input utterance by using the part of speech assignment process and one of the grammar rules that conclude to the grammatical relations of the input utterance.

Language Model. The implementation framework for constructing and using natural language interfaces has no previous knowledge of the schema of a repository, in general and there is a need to include all the possible synonyms and senses a word in user input may have. The synonyms that correspond to each sense of a word can be extracted by a thesaurus or a word ontology. We use WordNet [Miller, 1990] and in particular synsets and hyponyms for the linguistic analysis and the taxonomy of nouns.

Following the methodology and the algorithms described in the syntactic analysis procedure the natural language parser concludes to a language model described in figure 2. As we have already mentioned the object and its complements from dependent clauses are transformed into a new sentence without losing information, where the object becomes the subject. This structure contains information that helps the process of information retrieval with the subject and its complements, the object and its complements and for each word, its synsets and hyponyms from the WordNet.

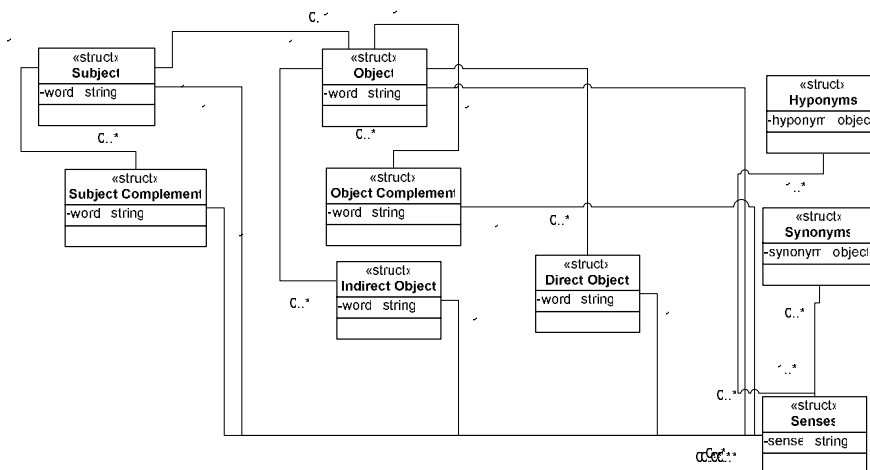


Figure 2. The language model that derives from the linguistic analysis

In this model diagram there are classes representing the grammatical relations that are connected with associations. What we see is that there are lists of words that constitute the basic sentence structures, like the subject and the object and there are complements and types of objects that predicate them.

2.3. Ontologies Processor

A domain can be described by a set of core and domain ontologies. It is crucial to identify more specific domains inside a domain, based on concepts and relationships of those concepts. For doing that we must define the degree of relatedness between pairs of related concepts. We call this procedure Semantic Similarity Clustering.

For capturing semantic similarity using ontologies that describe a domain, there is a need for a preprocessing phase to help and guide the process. It starts with tokenization of the multi-word terms of the ontology. The tokenization captures naming conventions used by database administrators, system integrators, programmers, etc. We also used domain-independent and domain-specific vocabularies like BABEL [http://www.ciw.uni-karlsruhe.de/kopien/babel.html], a glossary of Computer Oriented Abbreviations and Acronyms for abbreviation expansion. In the ontology preprocessing phase, we define the value of relatedness (similarity) between the concepts that describe the domain, by assigning weight values, based on the characteristics concerning the concepts and the relation types between this related concepts.

We have used cluster measure to compute the degree of similarity between two related concepts in a relation. The equation below describes the weight (similarity) between two related concepts, the percentage of concepts that C_k is related to, given that C_j is also related.

$$W(C_j, C_k) = \frac{\sum_{i=1}^n r_{ijk}}{\sum_{i=1}^n r_{ij}}$$

The value r_{ij} represents the fact that concept C_j is related to concept C_i (value: 0 or 1 in general). The value r_{ijk} represents the fact that both concepts C_j and C_k are related to concept C_i (value: 0 or 1 in general). The basic idea

of this measure is that concepts share more common relations with other concepts that are more similar. The benefit of using this measure is that this similarity measure is asymmetric. This is an important aspect for natural language processing since relations that are described with natural language do not indicate mathematical rules.

In OWL, classes provide an abstraction mechanism for grouping resources with similar characteristics. OWL contains three language constructs for combining class descriptions into class axioms, subClassOf, equivalentClass, and disjointWith. From these three constructs, the semantic relations that derive are: equivalence ($=$), more general (\supseteq), less general (\subseteq) and disjoint (\perp). The order of these relations, according to their binding strength, is as they have been listed, from the strongest to the weakest, except more and less general with those having the same binding power. These relations affect the weight assignment procedure as shown in Table 1.

We also make use of the class description of an OWL class, named property restriction. OWL distinguishes two kinds of property restrictions: value constraints and cardinality constraints. The cardinality constraint puts constraints on the number of values a property can take, in the context of this particular class description and can be related to the specificity value from Information Retrieval field, modeled as the idf (inverse domain frequency) measure [Yates and Neto, 1999]. In this case it denotes that the weight of a relation is inversely proportional to the number of relations with the concept that is the destination node of the relation. In conclusion, between related concepts C_j and C_k , the cardinality constraint n_k is equal to the number of instances of a given relation type that have k as its destination node.

$$W(C_j, C_k) = \frac{1}{\sqrt{n_k}}$$

Some more advanced class constructors of class descriptions are the intersection, the union and the complement that can be viewed as representing the AND, OR and NOT operators on classes.

Table 1. Listed parameters extracted from OWL Vocabulary and User Profile information that enhance semantically the mechanism for the weight assignment procedure $W(C_j, C_k)$ between related concepts

FACTOR	in concepts	W(Cj, Ck) values * $\in (0,1)$	New r values with User Profile information concerning a relation type when previous r values = 0.5	
			r_{ij}	r_{ij} and r_{ik}
Equivalence	$C_j = C_k$	*1	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$
More General	$C_j \supseteq C_k$	* $f_1 \in (0, 1)$	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$
Less General	$C_j \subseteq C_k$	* $f_1 \in (0, 1)$	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$
Disjoint	$C_j \perp C_k$	*0	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$
Intersection	$C_j \text{ AND } C_k$	* $f_2 \in (0, f_1)$	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$
Union	$C_j \text{ OR } C_k$	* $f_2 \in (0, f_1)$	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$
Complement	$C_j \text{ NOT } C_k$	* $f_2 \in (0, f_1)$	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$
No relation	C_j, C_k	* $f_3 \in (0, f_2)$	$0.5 \rightarrow 1$	$0.5 \rightarrow 1$

After listing the parameters that should be taken into account, we should specify the exact problem: how to create clusters of context inside one or more ontologies, describing the same domain, by calculating semantic relations between the concepts of the ontologies. This procedure averts an exhaustive search in the whole domain if we can eliminate the target of searching for the disambiguation.

The weight values that describe the relatedness between the concepts are extracted by taking into account the combined measure of the cluster measure and the specificity measure as they have presented previously. The specificity of a concept can be extracted by using the property – relation restriction, named cardinality constraint. Also, if there is information from any existing User Profile about relations that the user prefers, then a greater value of relatedness at the clustering procedure is assigned, because they are defining a preference of the user. This leads to a more personalized clustering of the domain. The similarity between two concepts can be determined, by using the cluster measure, but the values of the r_{ij} and r_{ik} will be determined after taking into account the semantics of OWL vocabulary and User Profile information. The percentage of influence of the weight values based on these factors it is still under consideration, since the experiments are still on going. In table 1 we list the parameters of influence and the impact that they have in the r values $\in \{0,0.5\}$ of the equation that follows were f_1, f_2, f_3 are multiplier factors to the product of cluster and specificity measure.

$$W(C_j, C_k) = \frac{\sum_{i=1}^n r_{ijk}}{\sum_{i=1}^n r_{ij}} * \frac{1}{\sqrt{n_k}}$$

2.4. Semantic Disambiguator

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 distin-

guishable from other meanings. Going a step further, the semantic disambiguation procedure is useful in applications where the user searches for concept instances of the model. The approach here is to obtain all concept instances (concepts in this phase and concept instances after the information retrieval from the repository) that are related to a given word even if that word does not appear inside the concept. The system can infer relations through techniques like spreading activation techniques on semantic networks [Crestani, 1997], since ontologies in OWL are described by semantic networks.

The constrained spreading activation algorithm is the main part of the semantic disambiguation procedure. From the weight assignment procedure, the relations between concepts have gained a numerical weight value. Considering an ontology as a concept graph, the spreading activation algorithm works by exploring this concept graph. The input of the algorithm is a set of concepts that refer to the user's query and respectively the concepts from the language model and the output is a set of strongly related concepts, that come up from navigation through linked concepts inside the graph.

The algorithm has as a starting point an initial set of concepts from the query (the nouns that contain the information to be searched in the information repository). The idea of the algorithm is to activate nodes that are semantically related to the initial nodes. The concepts that are extracted from the language model and constitute the initial set of the algorithm have an initial activation value based on their role in the request (subject, object, complement). The initial set is placed in a priority queue based on the node's initial weight value. The node with the highest activation value is the one processed. If it satisfies any constraints that may be applied it propagates its activation to the nodes that is related to (neighbors). If the initial node is i and the destination node is j , the propagation to the neighbor nodes is given by the following equation:

$$I_j(t) + = \sum_i O_i(t-1) * w_{ij} * (1 - \alpha)$$

- I_j is the total input of node j
- O_i is the total output of unit i connected to need j (in this system the output is equal to its input)
- w_{ij} is the weight associated to the relation connecting node i to node j , coming from the weight assignment procedure
- α is the percentage of activation that is lost every time an edge is processed (shortest paths better than longer paths).

The nodes that are activated and are not in the initial list are added to it, by an order specified from the weight values. The node that was processed is then inserted in the disambiguation list, which contains the nodes that were processed and are the result of the process. Nodes that are already in the disambiguation list can be extracted again for processing. This process continues till a predefined state (a defined output size for example) is reached or till there are no further nodes to be processed in the priority queue. At the end the disambiguation list contains all the nodes resulting from the constrained spreading activation algorithm ordered by their activation weight value.

The constraints applied, to limit the propagation from reaching the entire graph, are the distance constraint, the fan-out constraint (the spreading activation algorithm processes nodes with high connectivity to other nodes), the path constraint and the concept-type constraint.

3. Implementation of the OntoNL System

The logical architecture of the platform is depicted in Figure 4. From a data flow perspective, the user inserts a request to the Natural Language Parser using a java-based query interface, which does the complete syntactic and at a first level, semantic disambiguation, using the Stanford POS-Tagger, a syntactic analyzer, for applying grammar rules and rules for noun compound bracketing, and WordNet, as a complete word ontology. The output of the parser is a language model that needs to be fulfilled semantically, by the use of the ontologies. The module that takes as input ontologies does the preprocessing. It takes in input graphs codified into a standard OWL format. This module implements the preprocessing phase, the tokenization, the abbreviation expansion and the clustering based on weight assignment. The module responsible for the preprocessing of the ontology checks for any existing User Profiles and retrieves user preferences as concepts and relationships and enhances accordingly the measures for the weight assignment of related concepts of the domain ontology. The semantic disambiguation module applies a constrained spreading activation algorithm to the preprocessed graph representing the domain ontology/ies. This algorithm creates a list of one or more concept structures with information about instances of the concepts that a user may have declared. The semantic clustering of the ontologies is performed by applying a methodology that takes into account the strength of the relations between concepts, based on the syntax of OWL and User Profile information. The OWL ontologies that we used for a specific application are an upper OWL ontology fully capturing the MPEG-7 MDS [Tsinaraki et al., 2004] and a methodology for its extension with domain knowledge has been developed in the context of the DS-MIRF framework [Tsinaraki et al., 2003]. OWL/RDF metadata for audiovisual content description are produced, which are transformed, using appropriate transformation rules, to MPEG-7 compliant metadata, thus providing a basic level of MPEG-7 interoperability.

After the fully disambiguation of the user's request, the concept graphs are listed based on the most likely combination of concepts and instances for the information retrieval. Then, these graphs, via a Java API, are translated into XQueries that are applied to the MPEG-7 XML Repository. The MPEG-7 XML repository contains XML Documents, which are MPEG-7 compliant audiovisual content descriptions. XQueries return XML fragments that contain the requested information. A result formulation is needed, so the results to have a 'user-friendly' format. The platform's implementation is based on JSE 5.0 (figure 5). Other programming languages that were used are XML and OWL. We also used JWNL [<http://jwordnet.sourceforge.net/>], an API for accessing WordNet-style relational dictionaries, written in Java that also provides functionality beyond data access, such as relationship discovery and morphological processing.

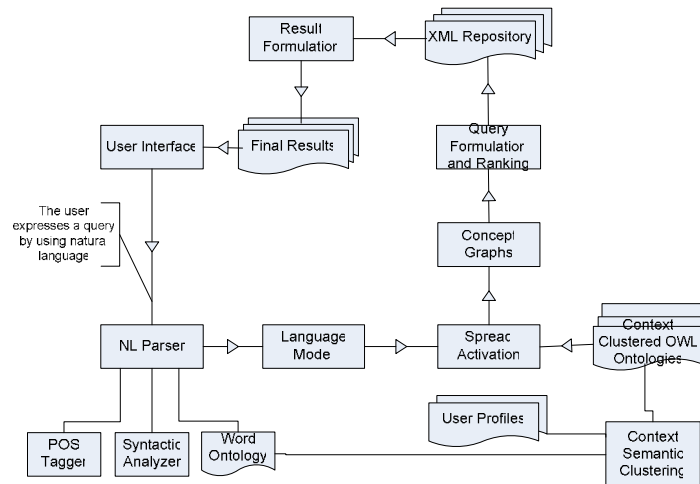


Figure 4. Architecture of the OntoNL system

We have created a demonstrator application that is comprised by a number of views that have the role to present the whole procedure followed from natural language to machine language and the retrieval of information. We can either load a file that contains a sentence or a text in the editor or either write our request. The converted sentence is then produced and contains the useful for the system information. The first part of the natural language parsing is the part-of-speech tagging that is based on the Stanford log-linear POS tagger.

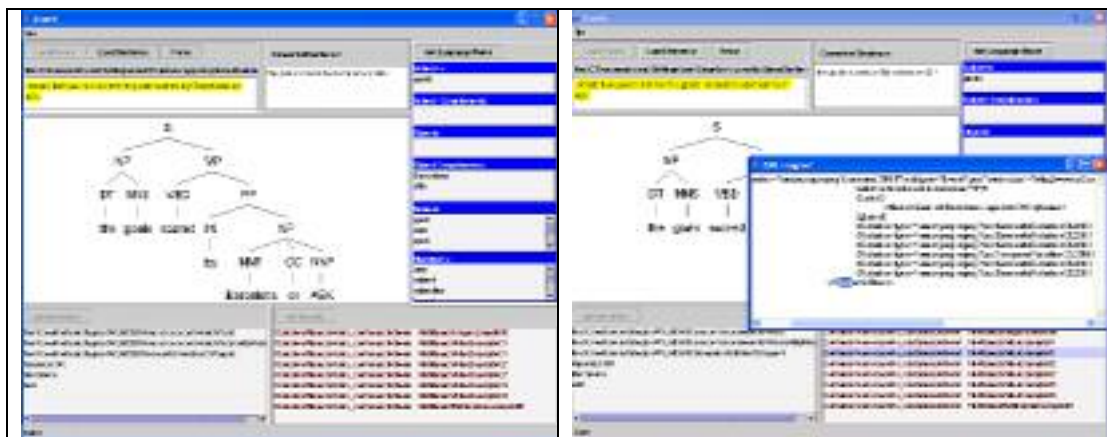


Figure 5. Screenshot of the OntoNL Framework for the domain of soccer.

In the right column we proceed with the linguistic analysis. We first check for noun compounds consulting the information from the ontologies and then we annotate the converted sentence with the grammatical relations. By this procedure we get the subjects, objects and their complements and by clicking in any of these words we retrieve its senses, synonyms and hyponyms from a word ontology, the WordNet. The lower part of the tool contains two views that concern domain information. The domain disambiguation view show us how the significant parts of the sentence are translated based on OWL domain ontologies. After this translation we get the results that are represented by a pair of values of the container and the id that shows where to find documents with the requested information. Then, by clicking we get the corresponding MPEG-7 XML fragment that contain the requested information.

4. Conclusions

We have presented in this report a framework for the automatic construction of Natural Language Interfaces to Multimedia Repositories (OntoNL System). The OntoNL system combines Natural Language Processing, with traditional search engine techniques and ontology-based information retrieval. OntoNL translates the user's requests to a language model that is enhanced with information coming from OWL ontologies of a specific domain.

The linguistic analysis phase includes a part-of-speech tagging, a noun compound analysis and a syntactic ('SVO') analysis. After this syntactic disambiguation, the system consults a word ontology (WordNet) for retrieving synonyms and senses of the words as a first stage of semantic disambiguation. The methodology used concludes to a language model, with syntactic relations of the words, senses, synonyms and hyponyms from the WordNet and part-of-speech taggers.

The domain specific ontologies are used by the system for semantic disambiguation. They are preprocessed to identify the strength with which concepts relate to each other. Strongly related concepts lead to higher ranked results during disambiguation. We presented a constrained spreading activation algorithm that was used for the retrieval of a list of the most relevant to the user's request concept structures that were translated into XQueries and addressed an MPEG-7 XML Repository.

The disambiguation procedure is automatic and quite promising, since it is linguistically as complete as possible in an automatic environment and it is enhanced with information based on the domain that the request refers to and user preference information. We believe that is easily reusable in many domains since the only restrictions are the English language from the Natural Language Processing part and OWL as the standard for representing ontologies of a specific domain.

The OntoNL functionality can be offered in a service-oriented fashion with Natural Language Interface generation for question-answering systems based on particular domains, with Search engine functionalities using either natural language either keywords, with keywords mapping in particular domain and query enhancement using personalized information (user profiles) and OWL ontologies and for MPEG-7 Repository Access (including search and filtering services).

We have shown the design and implementation of the framework. A detailed evaluation framework is currently produced, and experimentation with alternative algorithms continues. We hope to demonstrate in the long run that the implementation of semantic natural language interactions with multimedia repositories is feasible and inexpensive for a large number of applications domains and domain ontologies.

References

- Crestani, F. (1997). Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, 11(6): 453-482.
- Guarino, N. (1999). ONTOSEEK: Content-based access to the Web, *IEEE Intelligent Systems*, pp. 70-80
- Karanastasi, A., Kazasis, F., Christodoulakis, S. (2004). A Natural Language Model for Managing TV-Anytime Information from Mobile Devices. In *Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems (NLDB)*, Manchester, UK
- Karanastasi, A., Kazasis, F., Christodoulakis, S. (2005). A Natural Language Model and a System for Managing TV-Anytime Information in Mobile Environments, Special Issue on ACM Verlag International Journal of Personal and Ubiquitous Computing, Volume9, Number 5, 262-272
- Keller, F., Lapata, M. (2003). Using the Web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29:459-484.
- Lauer, M. (1995). Designing Statistical Language Learners: Experiments on Noun Compounds. Ph.D. thesis, Department of Computing Macquarie University NSW 2109 Australia.
- Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language*. MIT Press.
- Miller, G., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K.J. (1990). Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4), 235 -- 244
- Minnen, G., Carroll, J., Pearce, D. (2001). Applied morphological processing of English, *Natural Language Engineering*, 7(3). 207-223
- Nakov, P., Schwartz, A., Wolf, B., Hearst, M. (2005). Scaling up BioNLP: Application of a text annotation architecture to noun compound bracketing. In *Proceedings of SIG BioLINK*.
- Quirk, R., Greenbaum, S. (1973). *A University Grammar of English*. Longman
- Ratnaparkhi, A. (1996) A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing, EMNLP*, Pennsylvania, May, 1996
- Resnik, P. (1993). Selection and information: a class-based approach to lexical relationships. Ph.D. thesis, University of Pennsylvania, UMI Order No. GAX94-13894.
- Salembier, P. (2001) MPEG-7 Multimedia Description Schemes, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 6, 2001
- Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y. (2002). Managing Semantic Content for the Web. *IEEE Internet Computing* 6(4): 80-87
- 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 Technology*, Edmonton, Canada.
- Tsinaraki, C., Fatourou, E., Christodoulakis, S. (2003). An Ontology-Driven Framework for the Management of Semantic Metadata describing Audiovisual Information. In *Proceedings of CAiSE*, Velden, Austria, 2003, pp 340-356
- Tsinaraki, C., Polydoros, P., Christodoulakis, S. (2004). Integration of OWL Ontologies in MPEG-7 and TV-Anytime Compliant Semantic Indexing. In *Proceedings of CAiSE*: 398-413.
- Yates, B., Neto, B. (1999). *Modern Information Retrieval*. ACM Press, New York, USA, 1999.
- Vargas-Vera, M., Motta, E. (2004). AQUA – Ontology-based Question Answering System. *Third International Mexican Conference on Artificial Intelligence (MICAI-2004)*, Lecture Notes in Computer Science 2972 Springer Verlag.