Ontologically Supported Semantic Matching

Atanas K. Kiryakov, Kiril Iv. Simov
Linguistic Modelling Laboratory
Bulgarian Academy of Sciences
Acad. G. Bontchev Str. 25A, 1113 Sofia, Bulgaria
diogen@diogenes.bg, kivs@bgcict.acad.bg

Abstract

Evaluation of the closeness of two texts is a subtask for *FTR* and *IR* systems. The basic means used to accomplish it is the matching of *atomic text entities* (ATEs) such as words, stems, simple phrases and/or concepts. We address the question how concepts can be used as ATEs more efficiently in order to match *"small duck"* with *"small bird"*. The *onto-matching* technique introduced in the paper makes extensive use of lexical ontologies similar to WordNet.

We work with two tasks in mind: query expansion and text concept indexing. We outline some arguments showing why onto-matching is useful and how it can be implemented. Also, we conducted some experiments with query expansion for AltaVista.

1 Introduction

"A typical information retrieval task is to select documents from a database in response to a user's query, and rank these documents according to relevance." Strzalkowski et al (1998). The relevance must be defined on the basis of the concepts represented in the text and in the query. Usually information retrieval (IR) systems calculate the relevance of a text with respect to some query according to the number and the profile of the occurrences in the text of some elements from the query. The main stream of research in IR is towards the development of methods for the recognition of more meaning bearing elements of texts which can then be used to evaluate the closeness of the two texts (queries are also texts).

Most often a document is converted into a bag of words, stems or other textual elements which we call *atomic text entities* (ATEs) (sometimes information associated with them is also used). The hope is that these elements explicate the concepts represented by chunks of text and so define the topics of the document. Similarly, the query is considered to be itself a bag of words, stems, etc. and again the hope is that they explicate the concepts of the query.

Although words **denote** concepts, often they are not sufficient in themselves to **explicate** these concepts. They can be thought of as names for the concepts in the world. Usually the definition of a concept spells out what are the constraints on its possible representatives or instantiations, it could also give some prototype

information and information about this concept's relationship to other concepts. It is our opinion that users of information retrieval systems rarely search simply for words. Rather, they are interested in the concepts that words represent. Thus concepts (including at least some parts of their definitions and relations to other concepts) should be included amongst the atomic text entities. In this way we will capture the intuitive expectation that when one is searching for *bird* the occurrence of *duck* is also relevant. This is so because the word *duck* represents a subconcept (more specific concept) of the concept represented by *bird*.
The problem of the word-to-concept correspondence is well known and intensively studied in a number of areas like linguistics, psychology, artificial intelligence, etc. In order to demonstrate some of its aspects we give here a small example. Let us consider the following top-ontology of particulars (taken from Guarino (1998)):

```
Particular
    Location
        Space (a spatial region)
        Time (a temporal region)
    Object
        Concrete object
            Continuant (an apple)
            Occurrent (a fall of an apple)
        Abstract object (Pythagoras' theorem)
```

Here, objects are considered to be *concrete* because of their ability to have some location. *Continuants* are what is usually considered to be *objects*, while *occurrents* correspond to *events*. *Continuants* have a location in space. They have spatial parts, but they have neither a temporal location nor temporal parts. *Occurrents* are "generated" by continuants, according to the way they behave in time. Occurrents always have other occurrents as parts (continuants take occurrents as parts, but are not part of them). They have a unique temporal location, while their exact spatial location can not be defined in the general case. *Abstract objects* do not have a location at all. Most of the entities classified as abstract objects can also be thought of as *universals*.

Depending on the definition of a concept and therefore on the objects this concept denotes it can be classified under one or another branch of this ontology. Thus concepts *lexicalized* via words in a natural language (or *lexical concepts*) will belong to different branches of any ontology extending on the above minimal ontology. For example, the English word *book* denotes at least the following concepts: "information unit", "physical object" and "commodity" which belong to different branches. As a physical object *book* is a continuant and as an information unit it is an abstract object.

One another important point is that world knowledge, that is our repository of concepts and facts, is considerably more massive than is the set of lexicalized concepts. Therefore, if we use only words as concept denoting entities, we can hope to find only a fraction of the concepts that we have available to us.

In this paper we investigate the possibility to use WordNet (see Fellbaum (1998)) as a source for the explication of some concept relations in order to improve the matching of ATEs in documents and queries. More specifically, we exploit the hypernym-hyponym relation. We call this augmented matching of concepts — *onto-matching*. This improvement can be used in the core of both *FTR* and *IR* systems, as well as in other places, like information filtering, dictionary look up, information extraction, etc.

The structure of the paper is as follows: the following section gives an overview of WordNet and some of the approaches to using WordNet to enhance the precision in IR systems; afterwards, we discuss different approaches to "concept" search in texts and we introduce the central notion of the paper — onto-matching; the next section is devoted to the application of WordNet and onto-matching to query expansion and document indexing; the last section concludes the paper and lists some problems and directions for future research.

2 Using WordNet to Enhance IR

2.1 WordNet — a lexical ontology

The following is a concise description of WordNet as given by its developers: "WordNet is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets." see Miller (1995) and Fellbaum (1998). Some other basic relations are hyponymy, hypernymy and meronymy for nouns, entailment for verbs, antonymy for adjectives. The relations are divided in two levels: conceptual relations that connect synonym sets such as the hyponymy-hypernymy relation and lexical relations that connect particular words in synonym sets such as the antonymy relation.

The fundamental building block of WordNet — the *synonym set* or *synset* represents a lexical concept via the set of words (in some cases also phrases, idioms or collocations are used) that lexicalize this concept. Besides these "names" (and informal glosses) no other information is given about the concept, i.e. there are no formal definitions or prototype information. The main design principle of WordNet is to situate lexical concepts in semantic nets constructed with respect to a number of semantic relations between concepts (or words) that are sufficient to discriminate between them. This design principle implies the division of WordNet into four non-interacting semantic nets — one for each open word class. This separation is based on the fact that the appropriate semantic relations between the concepts represented by the synsets in the different parts of speech are incompatible. For example, there is no semantic relation that would appropriately connect a verb synset with a noun synset and that would make a reasonably detailed distinction between these synsets and other verb and/or noun synsets.

Additionally, the semantic nets in WordNet are divided in subnets by the so called unique beginners which determine hierarchies of mutually incomparable lexical concepts. These unique beginners play a role similar to that of the ontological classes given in the above top-ontology. The following synsets define some of the unique beginners for nouns:

{act, activity}	{animal, fauna}	{artifact}
{cognition, knowledge}	{natural object}	{possession}
{process}	{quantity, amount}	{shape}

The structure and the content of WordNet determine the ways and the extent to which it can be used to explicate the concepts in a text. Most profitably, one can use WordNet to determine the lexical concepts designated by a word and their relations to other lexical concepts.

2.2 WordNet and IR projects

WordNet was used in several projects to enhance the precision of the search for relevant documents. These include (among others): Voorhees (1998), Guarino et al (1999) and Gonzalo et al (1998). Gonzalo et al (1998) uses WordNet to index texts in two ways: first, they attach to each word its sense, using an index of three numbers — one for its part of speech, one for the unique beginner within this part of speech and a third one pointing to the word-sense in this file; second, they attach to each word the right synsets (lexical concepts). Then they use the standard vector based matching of the query to the documents using the added information. The experimental work shows that the performance of document retrieval by summaries improved by 29%!

Voorhees (1998) reported on two different tasks: word-sense disambiguation as part of the problem of conceptual matching and semantic expansion of the query. The conceptual matching experiment failed, because of a wrong strategy for automatic disambiguation combined with an extremely error-sensitive relevance evaluation method — the extended vector space model. The idea, in itself, is much like the one employed in Gonzalo et al (1998), where they studied the sensitivity of concept indexing against disambiguation errors and reported good results despite the 30% of errors. The goal of the second experiment was query expansion on the basis of lexical relations encoded in WordNet. All kinds of relations were studied as possible directions for the expansion with limited or unlimited transitivity. The results reported, however, concern only the case where all kinds of relations were traced for just one step. The conclusion was that such an expansion will lead to some improvements in the case of relatively short queries.

The OntoSeek project (Guarino et al (1999)) performs knowledge extraction with the support of the Sensus ontology (Knight & Luk (1994)). As a lexical front-end it uses WordNet and then maps the synsets to the formal concepts in Sensus. The goal is to provide means for knowledge acquisition from a knowledge base of lexical conceptual graphs (LCG). The target domains are on-line product catalogues and yellow pages. The results are descriptions of products or companies that can be matched with queries while taking into account ontological dependencies. This approach, however, presupposes semi-automatic encoding of the descriptions into a special form.

2.3 Including hypernymy in the retrieval

In our work we investigate the use of a more complicated concept matching approach augmenting some aspects of the approaches mentioned above. In our view a concept in a text is defined not only on the basis of its synset (taken as index) but also on the basis of other semantic relations, especially hyponymy and hypernymy.

We envisage two tasks: *Query expansion.* We expand the query by adding the hyponyms of the words it contains. Such an expanded query is evaluated with respect to documents for which no concept indexing has been done (using AltaVista for instance); *Concept indexing.* The texts of the documents are extended by a bag of concepts mapped to their words. These concepts are determined on the basis of the hypernymy-hyponymy relation.

3 Concept search

Concept search is defined in terms of atomic text entities which are extracted from texts and which are used as units in the evaluation of their closeness. In the usual query-document scenario we talk about query reengineering or expansion, while processing of the documents can be thought of as some sort of indexing. After the two texts (a document and a query) have been appropriately processed the sets of detected ATEs are matched with one another. There are also "stop-ATEs" which are defined in such a way that from each set of detected ATEs some of the entities are deleted. Usually, the deleted entities are those that denote overly broad concepts that would be found in any text. In our work we parametrize the notion of "stop-ATEs" to depend on the context and the wish of the user.

Most of the approaches to information retrieval we are aware of use the standard vector-space model to match the ATEs in texts and evaluate the semantic distance. The following is an overview of some of these approaches and the ATEs they use:

- *Word-stem.* With the help of an inflectional or a derivational morphological analyzer each word in the text is converted to its stem. The set of stems is used as an index space over which the matching algorithm operates. For instance, all occurrences of "read", "reads", "readable", "reader", "reading" are mapped to the stem "read". The idea is that a family of morphologically related words represents a concept and each member of the family denotes just some of aspect of it.

- *Word-sense.* This approach presupposes the availability of a lexical database listing a number of senses for each word. Using a word-sense disambiguator the appropriate sense is attached to each word in the text. The set of word-senses is used as an index space in the same way as in the word-stem approach. This approach is reported in Gonzalo et al (1998) where an index pointing to the word-sense is attached manually to the words in the test set of texts. The following example is taken from Gonzalo et al (1998): the occurrences of "debate" are represented by "debate%1:10:01::" where the three figures index is pointing to the sense number in the corresponding file of WordNet.

- *Lexical concept.* This approach uses a lexical database which relates each word to its corresponding lexical concepts. Each word in the text is substituted by an appropriate lexical concept. The index space here is the set of lexical concepts. In this approach different words can share the same lexical concept. For instance, in Gonzalo et al (1998) lexical concepts are represented by the synsets' identifiers from WordNet. Thus "debate" is substituted by "n04616654".

- *Ontological chunks.* Here the lexical concepts attached to the words in the text are augmented by their super and subconcepts. Thus each word is substituted by a chunk of an ontology which determines its place in it. Some of the ontological chunks will share their top parts — some lexical concepts in the text will have the same superconcepts. The index space is more complicated because we have to account for the ontological relations in the chunks. It is this approach that we investigate in this paper.

In the first three cases we can claim that the index spaces consist of points (word-stems, word-senses, lexical concepts). The matching algorithm has to compare these points in order to evaluate the matching of two texts. See Fig. 1 for a picture of this kind of matching. When the index space consists of ontological chunks, the matching algorithm has to be modified in an appropriate way to reflect the super/subconcept relation and the fact that concepts, even though they are not equivalent, could be considered relevant in the context of a certain retrieval task. This modification of the matching algorithm we will henceforth call **onto-matching**. See Fig. 2 for a picture of onto-matching.

The onto-matching approach is one attempt to overcome a certain intuitive asymmetry in users' expectations. For instance, in the case of *IR*, using the query/document schema, if one puts a more general query then all the documents that are evaluated as similar or more specific with respect to the concepts in the query are considered to be relevant. More general documents will be classified as irrelevant. When evaluating the relevance disregarding the structure of the texts, the same direction of generalization is expected for the atomic text entities (*ATEs*). For example, in most cases, a document containing only *bird* will be irrelevant to a query asking for *duck*. But under certain relevance evaluation schemata, a matching against the natural flow of generality can also be used (when in the document the superconcept is used in phrases that additionally constrain this superconcept and make it more specific).

Onto-matching gives more flexibility in the formulation of the query. The user can be additionally consulted so as to determine more exactly the content of the ontological chunks that are attached to the query. Depending on the settings, normally, the query will be indexed either by lexical concepts only or by lexical concepts and their subconcepts. In addition, one can direct the search using chunks that include also some of the neighboring concepts. This can be done by going up a few steps in the hierarchy to concept C and taking all subconcepts of C on the level of the lexical concept that was found in the text. Or to go on with our example, if in the text the concept *duck* is recognised, we go one step up in the hierarchy and take the immediate subconcepts. Then we also search for *goose* (this is done on the basis of

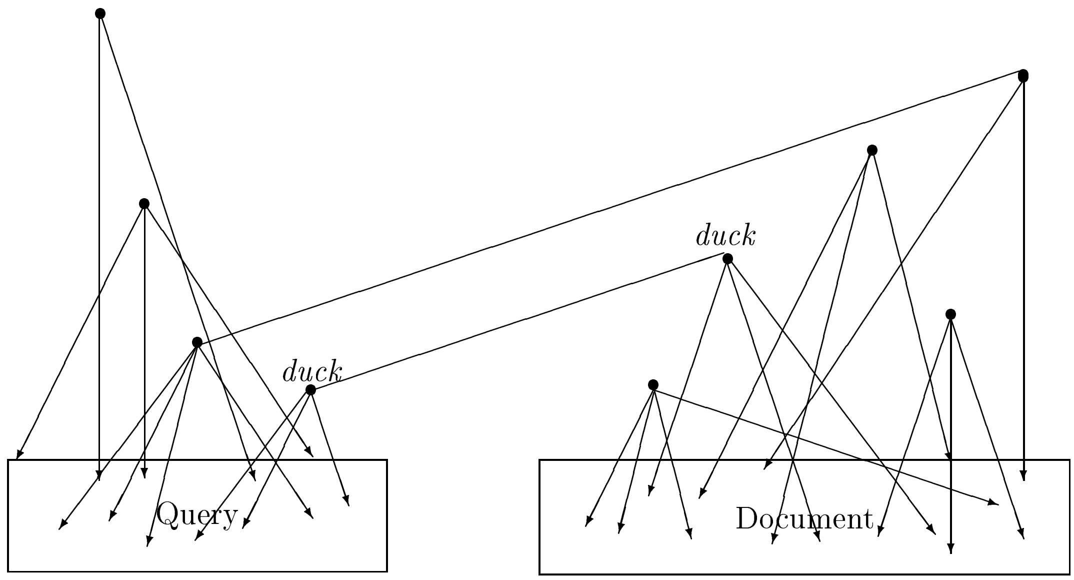

Fig. 1. Point to point matching. The dots represent the ATEs, the vectors represent the indexes to the occurrences of the ATEs in the text, the lines between the dots represent the matching between the ATEs in the query and in the document. For instance, the ATE for "duck" in the query matches the ATE for "duck" in the document.

the WordNet hierarchy). This can be done without the user's intervention if in the search engine an operator "SIMILAR" is defined that is doing this job automatically. Query expansion with ontological chunks can be useful also when the document collection with respect to which it will be evaluated is not indexed even by lexical concepts. In this case, we have to attach to the lexical concepts in the query their subconcepts (or the words that represent them). This approach can lead to generation of hundreds of alternatives just for one of the words in the query. For example, trying to get the transitive hyponym expansion of the synset for *bird* (the first one listed in WordNet), we will end up with more than one thousand synsets representing more specific concepts.

One way to control onto-matching is to "refine" and strip the ontological chunks by removing some of the concepts which are not relevant. Such judgement can be made because of irrelevancy to the users' goals in particular run. Or, because of the nature of onto-matching we might want to exclude some "artificial" concepts or other "noisy" patterns that can be recognized in the ontology. For example, if we are searching for *bird* it can be the case that we want to exclude some of the branches of the hyponyms like *seabird.* These refinements in onto-matching can not be made before the actual search because they depend on the users' goals.

The mechanisms proposed so far serve as a reduction of the problem of matching between relevant but non-equivalent concepts. The goal is to make possible onto-matching with minimal complication of the currently used algorithms. The ontological chunks explicitly represent the necessary inferences and they are taken into account by the standard matching algorithm.

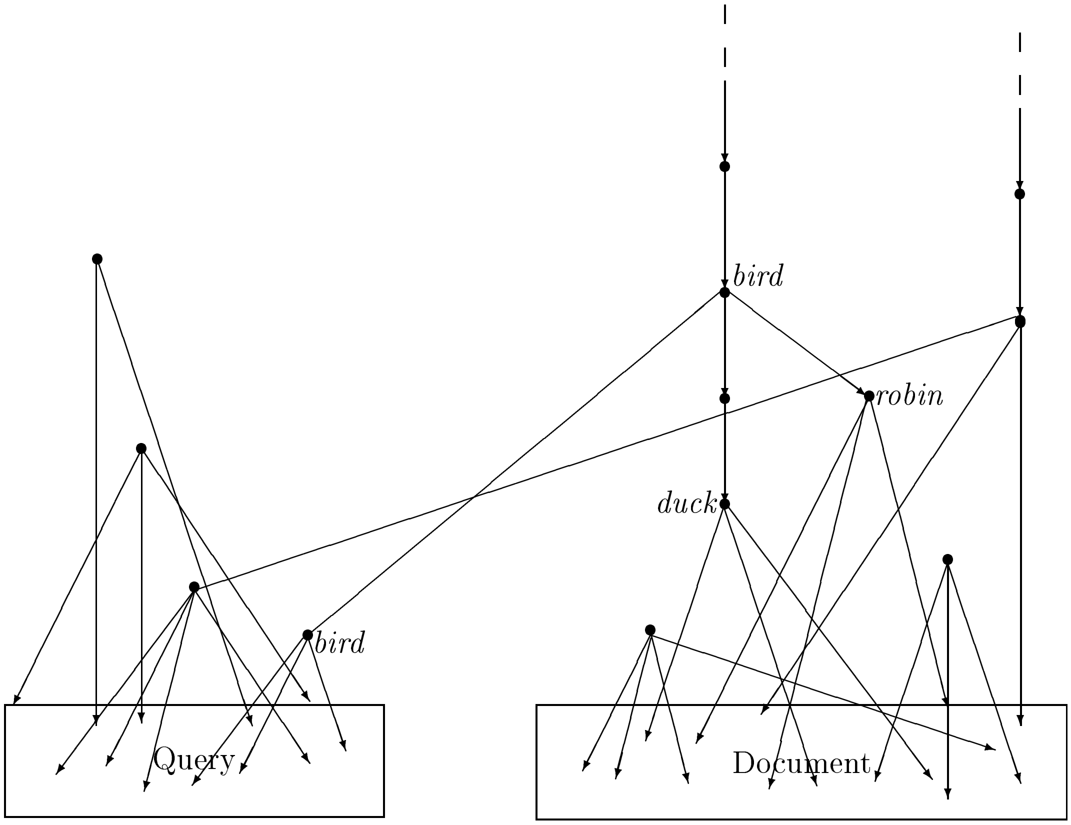

Fig. 2. Onto-matching. The dots represent the lexical concepts, the vectors represent the indexes to the occurrences of some lexical concepts in the text, the lines between the dots represent the matching between the lexical concepts in the query and in the document. Some lexical concepts are elements of the ontological chunks and the vectors connect them to their subconcepts instead of pointing directly to positions in the text. For instance, bird in the query matches duck and robin in the document via the ontological chunks above duck and robin.

4 WordNet for onto-matching

In this section we give some more concrete examples of the onto-matching approach using WordNet as the source for ontological chunks. We used the hypernymy-hyponymy relation between synsets for text indexing and query expansion. We conducted some experiments evaluating query expansion in searching the Internet. In both cases we presupposed that the texts were disambiguated and each word in them is connected to the right synset from WordNet. Some of the problems related to this assumption are commented in Section 5. In what follows we describe work with respect to the nouns in the text.

4.1 Text Indexing

Our goal is to index the text by the concepts corresponding to the words in it. Additionally, each lexical concept of a noun is indexed by the hypernym synsets.

All content words in the query text are indexed by their synsets only. Suppose a document contains an occurrence of the word *duck* to which the correct synset has already been assigned:

```
duck -- (small wild or domesticated web-footed broad-billed swimming bird ...)
```

The hypernyms for this synset are:

```
duck
   => anseriform bird
     => waterfowl, water bird, waterbird
       => aquatic bird
         => bird
           => vertebrate, craniate
             => chordate
               => animal, animate being, beast, brute, creature, fauna
                 => life form, organism, being, living thing
                   => entity, something
```

We index the occurrence of *duck* in the document with its synset and the synsets corresponding to its hypernyms. We call this **onto-indexing**. When a query contains a concept for *bird* it will be matched to the occurrence of *duck* (indexed with its hypernyms) without extending the query to the subconcept of *bird*.

One drawback of onto-indexing is that the overall size of the index will increase. We can partially solve this problem in two ways. First, we can reduce the number of the superconcepts deleting those that are not, strictly speaking, lexical concepts like "aquatic bird" in the hypernym chain above (see the Conclusion section). Also, the user can define concepts that should be excluded from the hypernym chain because they are specific to some domain of usage. In the example above one such lexical concept is "chordate" which is scientifically correct, but not much used in everyday life. Of course, if the search is for scientific documents then this concept should be retained and others excluded. Second, we can construct the ontology of the text so that hypernyms shared by some nouns in the text are represented only once. Suppose that in the text we have occurrences of *duck*, *goose* and *robin* and that *anseriform bird*, *aquatic bird* and *chordate* are excluded from the hypernym chains. In this case the index will look as follows:

$$
\begin{array}{c}
\{\text{entity, something}\} \\
\downarrow \\
\{\text{life form, organism, being, living thing}\} \\
\downarrow \\
\{\text{animal, animate being, beast, brute, creature, fauna}\} \\
\downarrow \\
\{\text{vertebrate, craniate}\} \\
\downarrow \\
\{\text{bird}\} \\
\swarrow \qquad \searrow \\
\begin{array}{cc}
\begin{array}{c}
\{\text{waterfowl, water bird, waterbird}\} \\
\swarrow \quad \searrow \\
\{\text{duck}\} \quad \{\text{goose}\}
\end{array}
&
\begin{array}{c}
\{\text{passerine, passeriform bird}\} \\
\downarrow \\
\{\text{oscine, oscine bird}\} \\
\downarrow \\
\{\text{thrush}\} \\
\downarrow \\
\{\text{robin}\}
\end{array}
\end{array}
\end{array}
$$

Thus reducing the number of the added superconcepts we hope that the increase of the index will be logarithmic to the size of the lexical concepts found in the text.

4.2 Query Expansion

Here we assume that a query is matched against a collection of documents that are indexed only by words or stems. We then use WordNet to generate a list of their synonyms and hyponyms. This list is added in an appropriate way to the original query and then the actual matching is done. This is the approach employed in our testbed, but we should mention that ambiguous words (synonyms or hyponyms) can lead to a sharp decline of precision.

We carried out some experiments with a query expansion in order to estimate the applicability of onto-matching for the retrieval of documents from a heterogeneous set (web-pages from AltaVista) with a short query. The query in this case is not a normal text but resembles a formula constructed from words and operators like `AND`, `OR`, `NEAR` and others. The words in the query were mapped manually to the correct synsets in WordNet. Then the full set of synonyms and hyponyms for each noun was constructed. This set was added to the query, with the exception of the multi-word phrases.
For instance, for the query "`+hotel NEAR +cheap NEAR +London`" we expanded the noun "hotel". The corresponding synset in WordNet is:

```
hotel -- (a building where travelers can pay for lodging
          and meals and other services)
```

This synset has the following hyponyms:

```
hotel
       => hostel, hostelry, inn, lodge
           => caravansary, caravanserai, khan, caravan inn
           => imaret
           => roadhouse
       => motel, motor hotel, motor inn, motor lodge, tourist court, court
       => resort hotel, spa
```

We added the hyponyms to the query connecting them to the expanded word with `OR`. In the expansion process we exclude the phrasal synonyms like "motor inn". After the expansion the query became:

```
   (hotel OR
            hostel OR hostelry OR inn OR lodge OR
                        caravansary OR caravanserai OR khan OR
                        imaret OR
                        roadhouse OR
            motel OR court OR
            spa)
    NEAR +cheap NEAR +London
```

AltaVista returned 104 documents for the original query, against 138 for the expanded one. Experiments concluded with different queries confirmed the expectation that the precision after the query expansion without onto-indexing is quite sensitive to ambiguous synonyms and hypernyms like "court" in the example. We concentrated on queries that do not contain highly ambiguous words in the expansion in order to get some approximation for the case of onto-matching.

We checked the precision for queries that return relatively small amount of documents and the general observation is that the expansion of the query did not depress it. The average increment of the recall is 30%.

In these experiments we were limited by practical considerations: we don't have at our disposal a collection of disambiguated documents indexed by lexical concepts; also, the majority of documents available for searches are not indexed by lexical concepts. Despite these practical constraints and the simplicity of the experiments, we can conclude that onto-matching is a promising approach for improving the precision in IR.

5 Conclusion

We found the results of the experiments encouraging, but have to point out a number of problems and directions for further research. The main difficulty is word-sense disambiguation. Throughout the paper we assume that each word in the text is correctly connected with the respective lexical concept. One could envisage a solution of this problem based on the use of semantic concordances in combination with statistical techniques similar to those used for POS-tagging.

Another problem is the recognition of multi-word concepts. For example, `water bird` is itself a concept and it will be strange to expand it in a query `bird` to something like:

```
water (bird OR cock OR hen OR eagle OR ...)
```

The right analyses of such terms will improve onto-indexing by attachment of the correct concept to multi-word terms in documents. This topic is described in details in Strzalkowski et al (1998).

If we have a more complicated concept representation where not only lexical concepts of nouns are used but also some additional constraints from the context are inferred (e.g. some attributes and their values) then a more sophisticated indexing mechanism will be needed. In this respect one can use the idea mentioned in Miller (1998) and compile for each noun a set of more appropriate attributes and their values (see pp. 40–41 there). These sets can be used for recognition of multi-word terms and for word-sense disambiguation.

In a more practical vein, we envisage to undertake experiments in onto-indexing over a collection of documents following the methodology of Gonzalo et al (1998) by manually attaching the appropriate chains of hypernyms to the words in the collection. These will give more reliable evidence for the usefulness of the ideas presented in this paper.

6 Acknowledgments

The research reported here was carried out within the Tübingen-Sofia International Graduate Programme in Computational Linguistics and Represented Knowledge (CLARK) funded by the Volkswagen-Stiftung. We would like to thank the Seminar für Sprachwissenschaft, Tübingen, for hosting the first phase of the writing of the paper. Also, we would like to thank Gergana Popova for her invaluable help. The first author is grateful to his colleagues from Sirma AI Ltd and Instill Corp. for their patience and understanding.

7 References

Gonzalo, J., Verdejo F., Chugur I. & Cigarran J. 1998. Indexing With WordNet Synsets Can Improve Text Retrieval. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*. Montreal.

Guarino, N. 1998. Some Ontological Principles for Designing Upper Level Lexical Resources. *Proceedings of First International Conference on Language Resources and Evaluation*. Granada, Spain.

Guarino, N., C. Masolo and Vetere G. 1999. OntoSeek: Using Large Linguistic Ontologies for Accessing On-Line Yellow Pages and Product Catalogs. *IEEE Intelligent Systems*.

Fellbaum, C. 1998. *WordNet: an electronic lexical database.* (editor) MIT Press.

Knight, K. & Luk S. 1994. Building a Large Knowledge Base for Machine Translation. *Proceeding American Association of Artificial Intelligence Conference (AAAI-94)*, AAAI Press, Menlo Park, California. 773-778.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of ACM*, 11, 39-41.

Miller, G. A. 1998. Nouns in WordNet. *WordNet: an electronic lexical database.* (editor) MIT Press.

Strzalkowski, T., Guthrie L., Karlgren J., Leistensnider J., Lin F., Perez-Carballo J., Straszheim T., Wang J. & Wilding J. 1998. *Natural Language Information Retrieval: TREC-5 Report.*

Voorhees, E. 1998. Using WordNet for Text Retrieval. *WordNet: an electronic lexical database.* (editor) MIT Press.