Semantic Indexing using WordNet Senses

Rada Mihalcea and Dan Moldovan
Department of Computer Science and Engineering
Southern Methodist University
Dallas, Texas, 75275-0122
{rada, moldovan}@seas.smu.edu

Abstract

We describe in this paper a boolean Information Retrieval system that adds word semantics to the classic word based indexing. Two of the main tasks of our system, namely the indexing and retrieval components, are using a combined word-based and sense-based approach. The key to our system is a methodology for building semantic representations of open text, at word and collocation level. This new technique, called *semantic indexing*, shows improved effectiveness over the classic word based indexing techniques.

1 Introduction

The main problem with the traditional boolean word-based approach to Information Retrieval (IR) is that it usually returns too many results or wrong results to be useful. Keywords have often multiple lexical functionalities (i.e. can have various parts of speech) or have several semantic senses. Also, relevant information can be missed by not specifying the exact keywords.

The solution is to include more information in the documents to be indexed, such as to enable a system to retrieve documents based on the words, regarded as lexical strings, or based on the semantic meaning of the words.

With this idea in mind, we designed an IR system which performs a combined word-based and sense-based indexing and retrieval.

The inputs to IR systems consist of a question/query and a set of documents from which the information has to be retrieved. We add lexical and semantic information to both the query and the documents, during a preprocessing phase in which the input question and the texts are disambiguated. The disambiguation process relies on contextual information, and identify the meaning of the words based on WordNet [1] (Fellbaum, 1998) senses. As described in the fourth section, we have opted for a disambiguation algorithm which is semi-complete (it disambiguates about 55% of the nouns and verbs), but is highly precise (over 92% accuracy), instead of using a complete but less precise disambiguation. A part of speech tag is also appended to each word. After adding these lexical and semantic tags to the words, the documents are ready to be indexed: the index is created using the words as lexical strings (to ensure a word-based retrieval), and the semantic tags (for the sense-based retrieval).

Once the index is created, an input query is answered using the document retrieval component of our system. First, the query is fully disambiguated; then, it is adapted to a specific format which incorporates semantic information, as found in the index, and uses the AND and OR operators implemented in the retrieval module.

Hence, using semantic indexing, we try to solve the two main problems of the IR systems described earlier. (1) relevant information is not missed by not specifying the exact keywords; with the new tags added to the words, we also retrieve words which are semantically related to the input keywords; (2) using the sense-based component of our retrieval sys-

[1] WordNet 1.6 is used in our system.

tem, the number of results returned from a search can be reduced, by specifying exactly the lexical functionality and/or the meaning of an input keyword.

The system was tested using the *Cranfield* standard test collection. This collection consists of 1400 documents, SGML formated, from the aerodynamics field. From the 225 questions associated with this data set, we have randomly selected 50 questions and build for each of them three types of queries: (1) a query that uses only keywords selected from the question, stemmed using the WordNet stemmer[2]; (2) a query that uses the keywords from the question and the synsets [3] for these keywords and (3) a query that uses the keywords from the question, the synsets for these keywords and the synsets for the keywords hypernyms. All these types of queries have been run against the semantic index described in this paper. Comparative results indicate the performance benefits of a retrieval system that uses a combined word-based and synset-based indexing and retrieval over the classic word based indexing.

2 Related Work

There are three main approaches reported in the literature regarding the incorporation of semantic information into IR systems: (1)*conceptual indexing*, (2) *query expansion* and (3) *semantic indexing*. The former is based on ontological taxonomies, while the last two make use of Word Sense Disambiguation algorithms.

2.1 Conceptual indexing

The usage of concepts for document indexing is a relatively new trend within the IR field. Concept matching is a technique that has been used in limited domains, like the legal field were conceptual indexing has been applied by (Stein, 1997). The FERRET system (Mauldin, 1991) is another example of how concept identification can improve IR systems.

To our knowledge, the most intensive work in this direction was performed by Woods (Woods, 1997), at Sun Microsystems Laboratories. He creates some custom built ontological taxonomies based on subsumtion and morphology for the purpose of indexing and retrieving documents. Comparing the performance of the system that uses conceptual indexing, with the performance obtained using classical retrieval techniques, resulted in an increased performance and recall. He defines also a new measure, called *success rate* which indicates if a question has an answer in the top ten documents returned by a retrieval system. The success rate obtained in the case of conceptual indexing was 60%, respect to a maximum of 45% obtained using other retrieval systems. This is a significant improvement and shows that semantics can have a strong impact on the effectiveness of IR systems.

The experiments described in (Woods, 1997) refer to small collections of text, as for example the Unix manual pages (about 10MB of text). But, as shown in (Ambroziak, 1997), this is not a limitation; conceptual indexing can be successfully applied to much larger text collections, and even used in Web browsing.

2.2 Query Expansion

Query expansion has been proved to have positive effects in retrieving relevant information (Lu and Keefer, 1994). The purpose of query extension can be either to broaden the set of documents retrieved or to increase the retrieval precision. In the former case, the query is expanded with terms similar with the words from the original query, while in the second case the expansion procedure adds completely new terms.

There are two main techniques used in expanding an original query. The first one considers the use of Machine Readable Dictionary; (Moldovan and Mihalcea, 2000) and (Voorhees, 1994) are making use of WordNet to enlarge the query such as it includes words

[2]WordNet stemmer = words are stemmed based on WordNet definitions (using the *morphstr* function)

[3]The words in WordNet are organized in *synonym sets*, called *synsets*. A synset is associated with a particular sense of a word, and thus we use *sense-based* and *synset-based* interchangeably.

which are semantically related to the concepts from the original query. The basic semantic relation used in their systems is the *synonymy* relation. This technique requires the disambiguation of the words in the input query and it was reported that this method can be useful if the sense disambiguation is highly accurate.

The other technique for query expansion is to use *relevance feedback*, as used in SMART (Buckley et al., 1994).

2.3 Semantic indexing

The usage of word senses in the process of document indexing is a pretty much debated field of discussions. The basic idea is to index word meanings, rather than words taken as lexical strings. A survey of the efforts of incorporating WSD into IR is presented in (Sanderson, 2000). Experiments performed by different researchers led to various, sometime contradicting results. Nevertheless, the conclusion which can be drawn from all these experiments is that a highly accurate Word Sense Disambiguation algorithm is needed in order to obtain an increase in the performance of IR systems.

Ellen Voorhees (Voorhees, 1998) (Voorhees, 1999) tried to resolve word ambiguity in the collection of documents, as well as in the query, and then she compared the results obtained with the performance of a standard run. Even if she used different weighting schemes, the overall results have shown a degradation in IR effectiveness when word meanings were used for indexing. Still, as she pointed out, the precision of the WSD technique has a dramatic influence on these results. She states that a better WSD can lead to an increase in IR performance.

A rather "artificial" experiment in the same direction of semantic indexing is provided in (Sanderson, 1994). He uses pseudo-words to test the utility of disambiguation in IR. A pseudo-word is an artificially created ambiguous word, like for example "banana-door" (pseudo-words have been introduced for the first time in (Yarowsky, 1993), as means of testing WSD accuracy without the costs associated with the acquisition of sense tagged corpora). Different levels of ambiguity were introduced in the set of documents prior to indexing. The conclusion drawn was that WSD has little impact on IR performance, to the point that only a WSD algorithm with over 90% precision could help IR systems.

The reasons for the results obtained by Sanderson have been discussed in (Schutze and Pedersen, 1995). They argue that the usage of pseudo-words does not always provide an accurate measure of the effect of WSD over IR performance. It is shown that in the case of pseudo-words, high-frequency word types have the majority of senses of a pseudo-word, i.e. the word ambiguity is not realistically modeled. More than this, (Schutze and Pedersen, 1995) performed experiments which have shown that semantics can actually help retrieval performance. They reported an increase in precision of up to 7% when sense based indexing is used alone, and up to 14% for a combined word based and sense based indexing.

One of the largest studies regarding the applicability of word semantics to IR is reported by Krovetz (Krovetz and Croft, 1993), (Krovetz, 1997). When talking about word ambiguity, he collapses both the morphological and semantic aspects of ambiguity, and refers them as *polysemy* and *homonymy*. He shows that word senses should be used in addition to word based indexing, rather than indexing on word senses alone, basically because of the uncertainty involved in sense disambiguation. He had extensively studied the effect of lexical ambiguity over IR; the experiments described provide a clear indication that word meanings can improve the performance of a retrieval system.

(Gonzalo et al., 1998) performed experiments in sense based indexing: they used the SMART retrieval system and a manually disambiguated collection (Semcor). It turned out that indexing by synsets can increase recall up to 29% respect to word based indexing. Part of their experiments was the simulation of a WSD algorithm with error rates of 5%, 10%, 20%, 30% and 60%: they found that error rates of up to 10% do not substantially af-

fect precision, and a system with WSD errors below 30% still perform better than a standard run. The results of their experiments are encouraging, and proved that an accurate WSD algorithm can significantly help IR systems.

We propose here a system which tries to combine the benefits of word-based and synset-based indexing. Both words and synsets are indexed in the input text, and the retrieval is then performed using either one or both these sources of information. The key to our system is a WSD method for open text.

3 System Architecture

There are three main modules used by this system:

1. **Word Sense Disambiguation** (WSD) module, which performs a semi-complete but precise disambiguation of the words in the documents. Besides semantic information, this module also adds part of speech tags to each word and stems the word using the WordNet stemming algorithm. Every document in the input set of documents is processed with this module. The output is a new document in which each word is replaced with the new format

   $$Pos|Stem|POS|Offset$$

   where: *Pos* is the position of the word in the text; *Stem* is the stemmed form of the word; *POS* is the part of speech and *Offset* is the offset of the WordNet synset in which this word occurs.

   In the case when no sense is assigned by the WSD module or if the word cannot be found in WordNet, the last field is left empty.

2. **Indexing** module, which indexes the documents, after they are processed by the WSD module. From the new format of a word, as returned by the WSD function, the *Stem* and, separately, the *Offset*|*POS* are added to the index. This enables the retrieval of the words, regarded as lexical strings, or the retrieval of the synset of the words (this actually means the retrieval of the given sense of the word and its synonyms).

3. **Retrieval** module, which retrieves documents, based on an input query. As we are using a combined word-based and synset-based indexing, we can retrieve documents containing either (1) the input keywords, (2) the input keywords with an assigned sense or (3) synonyms of the input keywords.

4 Word Sense Disambiguation

As stated earlier, the WSD is performed for both the query and the documents from which we have to retrieve information.

The WSD algorithm used for this purpose is an iterative algorithm; it was for the first time presented in (Mihalcea and Moldovan, 2000). It determines, in a given text, a set of nouns and verbs which can be disambiguated with high precision. The semantic tagging is performed using the senses defined in WordNet.

In this section, we present the various methods used to identify the correct sense of a word. Then, we describe the main algorithm in which these procedures are invoked in an iterative manner.

PROCEDURE 1. This procedure identifies the proper nouns in the text, and marked them as having sense #1.

Example. ``Hudson'' is identified as a proper noun and marked with sense #1.

PROCEDURE 2. Identify the words having only one sense in WordNet (*monosemous* words). Mark them with sense #1.

Example. The noun `subcommittee` has one sense defined in WordNet. Thus, it is a *monosemous* word and can be marked as having sense #1.

PROCEDURE 3. For a given word W_i, at position i in the text, form two pairs, one with the word before W_i (pair W_{i-1}-W_i) and the other one with the word after W_i (pair W_i-W_{i+1}). Determiners or conjunctions cannot

be part of these pairs. Then, we extract all the occurrences of these pairs found within the semantic tagged corpus formed with the 179 texts from SemCor(Miller et al., 1993). If, in all the occurrences, the word W_i has only one sense #k, and the number of occurrences of this sense is larger than 3, then mark the word W_i as having sense #k.

Example. Consider the word `approval` in the text fragment ``committee approval of''. The pairs formed are ``committee approval'' and ``approval of''. No occurrences of the first pair are found in the corpus. Instead, there are four occurrences of the second pair, and in all these occurrences the sense of `approval` is sense #1. Thus, `approval` is marked with sense #1.

PROCEDURE 4. For a given noun N in the text, determine the *noun-context* of each of its senses. This *noun-context* is actually a list of nouns which can occur within the context of a given sense i of the noun N. In order to form the *noun-context* for every sense N_i, we are determining all the concepts in the hypernym synsets of N_i. Also, using SemCor, we determine all the nouns which occur within a window of 10 words respect to N_i.

All of these nouns, determined using WordNet and SemCor, constitute the *noun-context* of N_i. We can now calculate the number of common words between this *noun-context* and the original text in which the noun N is found.

Applying this procedure to all the senses of the noun N will provide us with an ordering over its possible senses. We pick up the sense i for the noun N which: (1) is in the top of this ordering and (2) has the distance to the next sense in this ordering larger than a given threshold.

Example. The word `diameter`, as it appears in the document 1340 from the Cranfield collection, has two senses. The common words found between the *noun-contexts* of its senses and the text are: for `diameter#1`: { property, hole, ratio } and for `diameter#2`: { form}. For this text, the threshold was set to 1, and thus we pick `diameter#1` as the correct sense (there is a difference larger than 1 between the number of nouns in the two sets).

PROCEDURE 5. Find words which are semantically connected to the already disambiguated words for which the connection distance is 0. The distance is computed based on the WordNet hierarchy; two words are semantically connected at a distance of 0 if they belong to the same synset.

Example. Consider these two words appearing in the text to be disambiguated: `authorize` and `clear`. The verb `authorize` is a monosemous word, and thus it is disambiguated with procedure 2. One of the senses of the verb `clear`, namely sense #4, appears in the same synset with `authorize#1`, and thus `clear` is marked as having sense #4.

PROCEDURE 6. Find words which are semantically connected, and for which the connection distance is 0. This procedure is weaker than procedure 5: none of the words considered by this procedure are already disambiguated. We have to consider all the senses of both words in order to determine whether or not the distance between them is 0, and this makes this procedure computationally intensive.

Example. For the words `measure` and `bill`, both of them ambiguous, this procedure tries to find two possible senses for these words, which are at a distance of 0, i.e. they belong to the same synset. The senses found are `measure#4` and `bill#1`, and thus the two words are marked with their corresponding senses.

PROCEDURE 7. Find words which are semantically connected to the already disambiguated words, and for which the connection distance is maximum 1. Again, the distance is computed based on the WordNet hierarchy; two words are semantically connected at a maximum distance of 1 if they are *synonyms* or they belong to a *hypernymy/hyponymy* relation.

Example. Consider the nouns `subcommittee` and `committee`. The first one is disambiguated with procedure 2, and thus it is marked with sense #1. The word `committee` with its sense #1 is semantically linked with the word `subcommittee` by a *hypernymy* relation. Hence, we semantically tag this word

with sense #1.

PROCEDURE 8. Find words which are semantically connected between them, and for which the connection distance is maximum 1. This procedure is similar with procedure 6: both words are ambiguous, and thus all their senses have to be considered in the process of finding the distance between them.

Example. The words `gift` and `donation` are both ambiguous. This procedure finds `gift` with sense #1 as being the hypernym of `donation`, also with sense #1. Therefore, both words are disambiguated and marked with their assigned senses.

The procedures presented above are applied iteratively. This allows us to identify a set of nouns and verbs which can be disambiguated with high precision. About 55% of the nouns and verbs are disambiguated with over 92% accuracy.

Algorithm

Step 1. Pre-process the text. This implies tokenization and part-of-speech tagging. The part-of-speech tagging task is performed with high accuracy using an improved version of Brill's tagger (Brill, 1992). At this step, we also identify the complex nominals, based on WordNet definitions. For example, the word sequence ``pipeline companies'' is found in WordNet and thus it is identified as a single concept. There is also a list of words which we do not attempt to disambiguate. These words are marked with a special flag to indicate that they should not be considered in the disambiguation process. So far, this list consists of three verbs: *be, have, do.*

Step 2. Initialize the Set of Disambiguated Words (SDW) with the empty set SDW={}. Initialize the Set of Ambiguous Words (SAW) with the set formed by all the nouns and verbs in the input text.

Step 3. Apply procedure 1. The named entities identified here are removed from SAW and added to SDW.

Step 4. Apply procedure 2. The monosemous words found here are removed from SAW and added to SDW.

Step 5. Apply procedure 3. This step allows us to disambiguate words based on their occurrence in the semantically tagged corpus. The words whose sense is identified with this procedure are removed from SAW and added to SDW.

Step 6. Apply procedure 4. This will identify a set of nouns which can be disambiguated based on their *noun-contexts.*

Step 7. Apply procedure 5. This procedure tries to identify a *synonymy* relation between the words from SAW and SDW. The words disambiguated are removed from SAW and added to SDW.

Step 8. Apply procedure 6. This step is different from the previous one, as the *synonymy* relation is sought among words in SAW (no SDW words involved). The words disambiguated are removed from SAW and added to SDW.

Step 9. Apply procedure 7. This step tries to identify words from SAW which are linked at a distance of maximum 1 with the words from SDW. Remove the words disambiguated from SAW and add them to SDW.

Step 10. Apply procedure 8. This procedure finds words from SAW connected at a distance of maximum 1. As in step 8, no words from SDW are involved. The words disambiguated are removed from SAW and added to SDW.

Results

To determine the accuracy and the recall of the disambiguation method presented here, we have performed tests on 6 randomly selected files from SemCor. The following files have been used: br-a01, br-a02, br-k01, br-k18, br-m02, br-r05. Each of these files was split into smaller files with a maximum of 15 lines each. This size limit is based on our observation that small contexts reduce the applicability of procedures 5-8, while large contexts become a source of errors. Thus, we have created a benchmark with 52 texts, on which we have tested the disambiguation method.

In table 1, we present the results obtained for these 52 texts. The first column indicates the file for which the results are presented. The average number of nouns and verbs considered by the disambiguation algorithm for each of these files is shown in the second col-

Table 1: Results for the WSD algorithm applied on 52 texts

File	No. words	Proc.1+2		Proc.3		Proc.4		Proc.5+6		Proc.7+8	
		No.	Acc.	No.	Acc.	No.	Acc.	No.	Acc.	No.	Acc.
br-a01	132	40	100%	43	99.7%	58.5	94.6%	63.8	92.7%	73.2	89.3%
br-a02	135	49	100%	52.5	98.5%	68.6	94%	75.2	92.4%	81.2	91.4%
br-k01	68.1	17.2	100%	23.3	99.7%	38.1	97.4%	40.3	97.4%	41.8	96.4%
br-k18	60.4	18.1	100%	20.7	99.1%	26.6	96.9%	27.8	95.3%	29.8	93.2%
br-m02	63	17.3	100%	20.3	98.1%	26.1	95%	26.8	94.9%	30.1	93.9%
br-r05	72.5	14.3	100%	16.6	98.1%	27	93.2%	30.2	91.5%	34.2	89.1%
AVERAGE	88.5	25.9	100%	29.4	98.8%	40.8	95.2%	44	94%	48.4	92.2%

umn. In columns 3 and 4, there are presented the average number of words disambiguated with procedures 1 and 2, and the accuracy obtained with these procedures. Column 5 and 6 present the average number of words disambiguated and the accuracy obtained after applying procedure 3 (cumulative results). The cumulative results obtained after applying procedures 3, 4 and 5, 6 and 7, are shown in columns 7 and 8, 9 and 10, respectively columns 10 and 11.

The novelty of this method consists of the fact that the disambiguation process is done in an iterative manner. Several procedures, described above, are applied such as to build a set of words which are disambiguated with high accuracy: 55% of the nouns and verbs are disambiguated with a precision of 92.22%.

The most important improvements which are expected to be achieved on the WSD problem are *precision* and *speed*. In the case of our approach to WSD, we can also talk about the need for an increased *recall*, meaning that we want to obtain a larger number of words which can be disambiguated in the input text.

The precision of more than 92% obtained during our experiments is very high, considering the fact that WordNet, which is the dictionary used for sense identification, is very fine grained and sometime the senses are very close to each other. The accuracy obtained is close to the precision achieved by humans in sense disambiguation.

5 Indexing and Retrieval

The indexing process takes a group of document files and produces a new index. Such things as unique document identifiers, proper SGML tags, and other artificial constructs are ignored. In the current version of the system, we are using only the AND and OR boolean operators. Future versions will consider the implementation of the NOT and NEAR operators.

The information obtained from the WSD module is used by the main indexing process, where the word stem and location are indexed along with the WordNet synset (if present). Collocations are indexed at each location that a member of the collocation occurs.

All elements of the document are indexed. This includes, but is not limited to, dates, numbers, document identifiers, the stemmed words, collocations, WordNet synsets (if available), and even those terms which other indexers consider stop words. The only items currently excluded from the index are punctuation marks which are not part of a word or collocation.

The benefit of this form of indexing is that documents may be retrieved using stemmed words, or using synset offsets. Using synset offset values has the added benefit of retrieving documents which do not contain the original stemmed word, but do contain synonyms of the original word.

The retrieval process is limited to the use of the Boolean operators AND and OR. There is an auxiliary front end to the retrieval engine which allows the user to enter a textual query, such as, *"What financial institutions are found along the banks of the Nile?"* The auxiliary front end will then use the WSD to disambiguate the query and build a Boolean query for the standard retrieval engine.

For the preceding example, the auxil-

iary front end would build the query: *(financial_institution OR 6003131|NN) AND (bank OR 6800223|NN) AND (Nile OR 6826174|NN)* where the numbers in the previous query represent the offsets of the synsets in which the words with their determined meaning occur.

Once a list of documents meeting the query requirements has been determined, the complete text of each matching document is retrieved and presented to the user.

6 An Example

Consider, for example, the following question: *"Has anyone investigated the effect of surface mass transfer on hypersonic viscous interactions?"*. The question processing involves part of speech tagging, stemming and word sense disambiguation.

The question becomes: *"Has anyone investigate|VB|535831 the effect|NN|7766144 of surface|NN|3447223 mass|NN|3923435 transfer|NN|132095 on hypersonic|JJ viscous|JJ interaction|NN|7840572"*.

The selection of the keywords is not an easy task, and it is performed using the set of 8 heuristics presented in (Moldovan et al., 1999). Because of space limitations, we are not going to detail here the heuristics and the algorithm used for keywords selection. The main idea is that an initial number of keywords is determined using a subset of these heuristics. If no documents are retrieved, more keywords are added, respectively a too large number of documents will imply that some of the keywords are dropped in the reversed order in which they have been entered.

For each question, three types of query are formed, using the AND and OR operators.

1. Q_{WNStem}. Keywords from the question, stemmed based on WordNet, concatenated with the AND operator.

2. $Q_{WNOffset}$. Keywords from the question, stemmed based on WordNet, concatenated using the OR operator with the associated synset offset, and concatenated with the AND operator among them.

3. $Q_{WNHyperOffset}$. Keywords from the question, stemmed based on WordNet, concatenated using the OR operator with the associated synset offset and with the offset of the hypernym synset, and concatenated with the AND operator among them.

All these types of queries are run against the semantic index created based on words and synset offsets. We denote these runs with R_{WNStem}, $R_{WNOffset}$ and $R_{WNHyperOffset}$.

The three query formats, for the given question, are presented below:

Q_{WNstem}. *effect* AND *surface* AND *mass* AND *flow* AND *interaction*

$Q_{WNOffset}$. (*effect* OR *7766144|NN*) AND (*surface* OR *3447223|NN*) AND (*mass* OR *3923435|NN*) AND (*transfer* OR *132095|NN*) AND (*interaction* OR *7840572|NN*)

$Q_{WNHyperOffset}$ (*effect* OR *7766144|NN* OR *20461|NN*) AND (*surface* OR *3447223|NN* OR *11937|NN*) AND (*mass* OR *3923435|NN* OR *3912591|NN*) AND (*transfer* OR *132095|NN* OR *130470|NN*) AND (*interaction* OR *7840572|NN* OR *7770957|NN*)

Using the first type of query, 7 documents were found out of which 1 was considered to be relevant. With the second and third types of query, we obtained 11, respectively 17 documents, out of which 4 were found relevant, and actually contained the answer to the question.

(sample answer) ... *the present report gives an account of the development of an approximate theory to the problem of hypersonic strong viscous interaction on a flat plate with mass-transfer at the plate surface. the disturbance flow region is divided into inviscid and viscous flow regions* .... (cranfield0305).

7 Results

The system was tested on the *Cranfield* collection, including 1400 documents, SGML formated[4]. From the 225 questions provided

[4]Demo available online at http://pdp13.seas.smu.edu/ rada/sem.ind./

with this collection, we randomly selected 50 questions and used them to create a benchmark against which we have performed the three runs described in the previous sections: R_{WNStem}, $R_{WNOffset}$ and $R_{WNHyperOffset}$.

For each of these questions, the system forms three types of queries, as described above. Below, we present 10 of these questions and show the results obtained in Table 2.

1. Has anyone investigated the effect of surface mass transfer on hypersonic viscous interactions?

2. What is the combined effect of surface heat and mass transfer on hypersonic flow?

3. What are the existing solutions for hypersonic viscous interactions over an insulated flat plate?

4. What controls leading-edge attachment at transonic velocities?

5. What are wind-tunnel corrections for a two-dimensional aerofoil mounted off-centre in a tunnel?

6. What is the present state of the theory of quasi-conical flows?

7. References on the methods available for accurately estimating aerodynamic heat transfer to conical bodies for both laminar and turbulent flow.

8. What parameters can seriously influence natural transition from laminar to turbulent flow on a model in a wind tunnel?

9. Can a satisfactory experimental technique be developed for measuring oscillatory derivatives on slender sting-mounted models in supersonic wind tunnels?

10. Recent data on shock-induced boundary-layer separation.

Three measures are used in the evaluation of the system performance: (1) *precision*, defined as the number of relevant documents retrieved over the total number of documents retrieved; (2) *recall*, defined as the number of relevant documents retrieved over the total number of relevant documents found in the collection and (3) *F-measure*, which combines both the precision and recall into a single formula:

$$F_{measure} = \frac{(\beta^2 + 1.0) * P * R}{(\beta^2 * P) + R}$$

where P is the precision, R is the recall and β is the relative importance given to recall over precision. In our case, we consider both precision and recall of equal importance, and thus the factor β in our evaluation is 1.

The tests over the entire set of 50 questions led to 0.22 precision and 0.25 recall when the WordNet stemmer is used, 0.23 precision and 0.29 recall when using a combined word-based and synset-based indexing. The usage of hypernym synsets led to a recall of 0.32 and a precision of 0.21.

The relative gain of the combined word-based and synset-based indexing respect to the basic word-based indexing was 16% increase in recall and 4% increase in precision. When using the hypernym synsets, there is a 28% increase in recall, with a 9% decrease in precision.

The conclusion of these experiments is that indexing by synsets, in addition to the classic word-based indexing, can actually improve IR effectiveness. More than that, this is the first time to our knowledge when a WSD algorithm for open text was actually used to automatically disambiguate a collection of texts prior to indexing, with a disambiguation accuracy high enough to actually increase the recall and precision of an IR system.

An issue which can be raised here is the efficiency of such a system: we have introduced a WSD stage into the classic IR process and it is well known that WSD algorithms are usually computationally intensive; on the other side, the disambiguation of a text collection is a process which can be highly parallelized, and thus this does not constitute a problem anymore.

8 Conclusions

The full understanding of text is still an elusive goal. Short of that, semantic indexing offers an improvement over current IR techniques. The key to semantic indexing is fast WSD of large collections of documents.

In this paper we offer a WSD method for open domains that is fast and accurate. Since only 55% of the words can be disambiguated so far, we use a hybrid indexing approach that combines word-based and sense-based indexing. The senses in WordNet are fine grain and the WSD method has to cope with this. The

Table 2: Results for 10 questions run against the three indices created on the *Cranfield* collection. The bottom line shows the results for the entire set of questions.

Question number	Query type								
	R_{WNStem}			$R_{WNOffset}$			$R_{WNHyperOffset}$		
	recall	precision	f-measure	recall	precision	f-measure	recall	precision	f-measure
1	0.08	0.14	0.05	0.31	0.36	0.17	0.31	0.24	0.14
2	0.06	0.17	0.04	0.25	0.44	0.16	0.25	0.31	0.14
3	0.47	0.70	0.28	0.47	0.70	0.28	0.53	0.67	0.30
4	0.25	0.60	0.18	0.25	0.60	0.18	0.25	0.60	0.18
5	0.33	0.50	0.20	1.00	0.25	0.20	1.00	0.19	0.16
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.17	0.17	0.09	0.17	0.17	0.09	0.17	0.17	0.09
8	0.20	0.11	0.07	0.20	0.11	0.07	0.20	0.11	0.07
9	0.67	0.50	0.29	0.67	0.50	0.29	1.00	0.38	0.28
10	0.29	0.07	0.06	0.29	0.07	0.06	0.29	0.06	0.05
AVG/50	0.25	0.22	0.09	0.29	0.23	0.11	0.32	0.21	0.10

WSD algorithm presented here is new for the NLP community and proves to be well suited for a task such as semantic indexing.

The continuously increasing amount of information available today requires more and more sophisticated IR techniques, and semantic indexing is one of the new trends when trying to improve IR effectiveness. With semantic indexing, the search may be expanded to other forms of semantically related concepts as done by Woods (Woods, 1997). Finally, semantic indexing can have an impact on the semantic Web technology that is under consideration (Hellman, 1999).

References

J. Ambroziak. 1997. Conceptually assisted Web browsing. In *Sixth International World Wide Web conference*, Santa Clara, CA. full paper available online at http://www.scope.gmd.de/info/www6/posters/ 702/guide2.html.

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, Trento, Italy.

C. Buckley, G. Salton, J. Allan, and A. Singhal. 1994. Automatic query expansion using smart: Trec 3. In *Proceedings of the Text REtrieval Conference (TREC-3)*, pages 69–81.

C. Fellbaum. 1998. *WordNet, An Electronic Lexical Database*. The MIT Press.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. 1998. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of COLING-ACL '98 Workshop on Usage of WordNet in Natural Language Processing Systems*, Montreal, Canada, August.

R. Hellman. 1999. A semantic approach adds meaning to the Web. *Computer*, pages 13–16.

R. Krovetz and W.B. Croft. 1993. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141.

R. Krovetz. 1997. Homonymy and polysemy in information retrieval. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 72–79.

X.A. Lu and R.B. Keefer. 1994. Query expansion/reduction and its impact on retrieval effectiveness. In *The Text REtrieval Conference (TREC-3)*, pages 231–240.

M.L. Mauldin. 1991. Retrieval performance in FERRET: a conceptual information retrieval system. In *Proceedings of the 14th International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 347–355, Chicago, IL, October.

R. Mihalcea and D.I. Moldovan. 2000. An iterative approach to word sense disambiguation. In *Proceedings of FLAIRS-2000*, pages 219–223, Orlando, FL, May.

G. Miller, C. Leacock, T. Randee, and R. Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, New Jersey.

D Moldovan and R. Mihalcea. 2000. Using WordNet and lexical operators to improve Internet searches. *IEEE Internet Computing*, 4(1):34–43.

D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Girju, and V. Rus. 1999. LASSO: A tool for surfing the answer net. In *Proceedings of the Text Retrieval Conference (TREC-8)*, November.

M. Sanderson. 1994. Word sense disambiguation and information retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 142–151, Springer-Verlag.

M. Sanderson. 2000. Retrieving with good sense. *Information Retrieval*, 2(1):49–69.

H. Schutze and J. Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

J.A. Stein. 1997. Alternative methods of indexing legal material: Development of a conceptual index. In *Proceedings of the Conference "Law Via the Internet 97"*, Sydney, Australia.

E.M. Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR, Conference on Research and Development in Information Retrieval*, pages 61–69, Dublin, Ireland.

E.M. Voorhees. 1998. Using WordNet for text retrieval. In *WordNet, An Electronic Lexical Database*, pages 285–303. The MIT Press.

E.M. Voorhees. 1999. Natural language processing and information retrieval. In *Information Extraction: towards scalable, adaptable systems. Lecture notes in Artificial Intelligence, #1714*, pages 32–48.

W.A. Woods. 1997. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, April. available online at: *http://www.sun.com/ research/techrep/ 1997/abstract-61.html.*

D. Yarowsky. 1993. One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop.*