

Improving the search on the Internet by using WordNet and lexical operators

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

June 30, 1998

Not for distribution or attribution. For review purposes only.

Abstract

A vast amount of information is available on the Internet, and naturally, many information gathering tools have been developed. Search engines with different characteristics, such as AltaVista, Lycos, Infoseek, and others are available. However, there are inherent difficulties associated with the task of retrieving information on the Internet: (1) the web information is diverse and highly unstructured, (2) the size of information is large and it grows at an exponential rate. While these two issues are profound and require long term solutions, still it is possible to develop software around the search engines to improve the quality of the information retrieved. In this paper we present a natural language interface system to a search engine. The search improvement achieved by our system is based on: (1) a query extension using WordNet and (2) the use of new lexical operators that replace the classical boolean operators used by current search engines. Several tests have been performed using the TIPSTER topics collection, provided at the 6th Text Retrieval Conference (TREC-6); the results obtained are presented and discussed.

1 Introduction

A main problem with the current search engines is the large volume of documents extracted as a result of broad, general queries, and the lack of output produced to specific, narrow questions [Selberg and Etzioni 1995], [Zorn, Emanoil et al. 1996]. Many of the documents retrieved for general queries are totally irrelevant to the subject of interest and relevant documents may be missing because the query does not contain the exact keywords. A more refined query, with more restrictive boolean operators, may result in a few or even no documents.

The performance of a search engine is measured based on both the relevance of the information retrieved and the number of relevant documents retrieved. The evaluation methodology is based on two factors: the *precision* and the *recall*. The *precision* is defined as the ratio between the number of relevant documents retrieved over the total number of documents retrieved. The *recall* is defined as the ratio between the number of relevant documents

extracted over the total number of relevant documents in the database. Increasing one of these factors usually decreases the other one, and vice-versa.

Two main approaches have been considered by researchers in trying to improve the quality of the search on Internet or large collections of texts.

The first one is to make use of multiple search engines and create a meta search engine [Selberg and Etzioni 1995], [Gravano, Chang et al. 1997]. This will result in an increased number of documents, as they are retrieved based on the information stored in multiple search engine databases. The hard task in this approach is that different search engines are largely incompatible and do not always allow for interoperability. Solving this problem implies a unification of both the query language and the type of results returned by the different search engines.

The second approach is to use Natural Language Processing (NLP) techniques. Here, work has been developed in two directions. (1) Machine Readable Dictionaries have been used for query extension to increase the number of documents retrieved. This kind of methods has been developed for information retrieval on the Internet [Allen 1997] or in very large text collections [Voorhees 1994]. (2) Improve the quality of the information retrieved using NLP-based systems: REASON [Anikina, Golender et al. 1997], INQUIRY [Callan, Croft et al. 1992].

These techniques have been designed to increase one of the two evaluation factors described above: the *recall* or the *precision*. It is known that *recall* and *precision* tend to vary inversely and that it is difficult to retrieve everything that is wanted, while rejecting everything that is unwanted. Still, it is possible to use a more sophisticated approach that “serially” combines these methods: increase first the recall, and than for the larger set of documents obtained, increase the quality of the information retrieved.

An aspect that needs attention is to bridge the gap between the human questions and the simple query format that search engines take. When we want information, we think in terms of questions that are far more complex than simple words or combinations of words currently accepted by the search engines. One may ask:

Q1: ‘‘Who were the US Presidents of the last century?’’, or

Q2: ‘‘I want to know who was the 10th President of the United States, his religion and where was he borne’’.

This calls for a natural language interface that transforms sentences into queries with boolean operators currently accepted by the search engines. For many applications, such an interface does not have to perform a complex parsing and a deep semantic analysis of the input sentence. It may be sufficient to recognize the main concepts by performing a shallow linguistic processing. However, one thing that is highly beneficial is to search not only for the words that occur in the input sentence, but to create **similarity lists** with words from on-line dictionaries that have the same meaning as the input words. This can significantly broaden the web search.

Another area of improvement may be to design new retrieval operators that retain more relevant documents. A possible approach along this direction is still to use the existent search engines, which were developed with great efforts, but to post-process the set of documents produced to a query.

In this paper we examine some of the benefits of using Natural Language Processing in conjunction with WordNet, an on-line lexical database developed at Princeton University [Miller 1995], to improve the quality of the results. Specifically, the paper describes a system that addresses two issues: (1) the translation of a natural language question or sentence into

a query and query extension using WordNet, and (2) extraction of paragraphs that render relevant information from the documents fetched by the search engines. The first step is intended to increase the *recall* while the second one increases the *precision*.

2 Background on resources

Different resources have been used in developing and testing the system described in this paper. The first task performed by the system, namely the translation of a natural language question into a query and then query expansion, is done using WordNet. The second task, i.e. fetching documents from the Internet and information extraction, makes use of the AltaVista search engine. The system has been tested using 50 questions derived from the topics provided at the 6th Text Retrieval Conference (TREC-6).

2.1 AltaVista

AltaVista [AltaVista] is a search engine developed in 1995 by the Digital Equipment Corporation in its Palo Alto research labs. There are several characteristics of this search service that makes AltaVista one of the most powerful search engines. In choosing AltaVista for use in our system, we based our decision on two of these features: (1) the size of information on Internet that can be accessed through AltaVista: it has a growing index of over 125,000,000 unique World Wide Web pages; (2) it accepts complex boolean searches through its *advanced search* function. These features make this search engine suitable for the development of software around it, with the goal of increasing the quality of the information retrieved.

Specific relationships can be created among the keywords of a query accepted by AltaVista. These relations can be created using brackets, *AND*, *OR*, *NOT* and *NEAR* operators. *AND* finds only documents containing all of the specified words or phrases. *Mary AND lamb* finds documents with both the word *Mary* and the word *lamb*. *OR* finds documents containing at least one of the specified words or phrases. *Mary OR lamb* finds documents containing either *Mary* or *lamb*. The documents retrieved may contain both words, but not necessarily. *NEAR* finds documents containing both specified words or phrases within 10 words of each other. *Mary NEAR lamb* finds documents containing both the word *Mary* and the word *lamb* but with the restriction that these words are separated by maximum 10 other words.¹

2.2 WordNet

WordNet² is a Machine Readable Dictionary developed at Princeton University by a group led by George Miller [Miller 1995]. It is used by our system for word sense disambiguation and generation of similarity lists.

WordNet covers the vast majority of nouns, verbs, adjectives and adverbs from the English language. The words in WordNet are organized in synonym sets, called *synsets*. Each synset represents a concept. There is a rich set of relation links between words and other words, between words and synsets, and between synsets; these lexico-semantic relations make WordNet a useful resource for natural language processing.

¹These examples are from the *AltaVista Advanced Help*
http://www.altavista.digital.com/av/content/help_advanced.htm

²WordNet 1.5 has been used in our algorithm implementation.

The version 1.5 contains a large network of 168,217 words, organized in 91,595 synssets, connected through 345,264 semantic links (Table 1).

Part of speech	words	concepts
noun	107,884	60,557
verb	25,768	11,363
adjective	28,762	16,428
adverb	6,203	3,243
Total	168,217	91,591

Table 1: The number of words and concepts in WordNet 1.5

The noun concepts are spanned by 11 *isa* hierarchies, whereas the verb concepts are classified into 558 *isa* hierarchies. For almost all the synssets, a gloss is also provided, including an explanation of the words within the synset and an example.

2.3 TREC topics

The Text Retrieval Conferences (TREC) are part of the TIPSTER Program, and are intended to encourage research in information retrieval from large texts. The information needs are described by data structures called *topics*.

The TIPSTER project distinguishes between two different types of queries: *ad hoc* and *routing*. The *ad hoc* queries are designed to investigate the performance of systems that search a set of documents using novel topics; these are most suitable for systems implying specific searches. The *routing* queries investigate the performance of systems that use standing queries to search new streams of documents; the systems using this task usually address general searches; a routing query can be viewed as a filter on incoming documents.

As our interface is designed to improve the quality of the information retrieved especially in the case of specific questions, we used the *ad hoc* topics in order to test the performance of our system. We derived 50 natural language questions from the ad hoc topics provided at the 6th Text Retrieval Conferences [TREC 1997].

An example of a topic from the TREC-6 *ad hoc* collection is presented in Figure 1. As it can be seen from this figure, a topic is a frame-like data structure. Its fields are to be interpreted as follows: the `<num>` section identifies the topic; the `<title>` section classifies the topic within a domain; the `<desc>` section gives a brief description of the topic (for TREC-6, this section was intended to be an initial search query); the `<narr>` section provides a further explanation of what a relevant material may look like.

```

<num> Number: 301
<title> International Organized Crime
<desc> Description:
Identify organization that participate in international criminal activity, the activity, and, if possible, collaborating
organization and the countries involved.
<narr> Narrative:
A relevant document must as a minimum identify the organization and the type of illegal activity (e.g., Columbian
cartel exporting cocaine). Vague references to international drug trade without identification of the organization(s)
involved would not be relevant.

```

Figure 1: A TIPSTER topic

For the purpose of testing our system, we used the *<desc>* field to derive natural language questions in a form similar with the questions normally used by users to search the Internet. For example, from the corpus entry presented above, the question that we derived is: ‘‘Which are some of the organizations participating in international criminal activity?’’ .

After retrieving the information using the derived questions, the relevance of the information has been evaluated based on the narrative section of each topic.

3 System architecture

The system architecture is shown in Figure 2. An input query or sentence expressed in English is first presented to the lexical processing module that extracts the keywords from the sentence. The query formation module uses these keywords to form queries that are sent to one or more search engines. The documents fetched by the search engines are filtered with the help of some new operators.

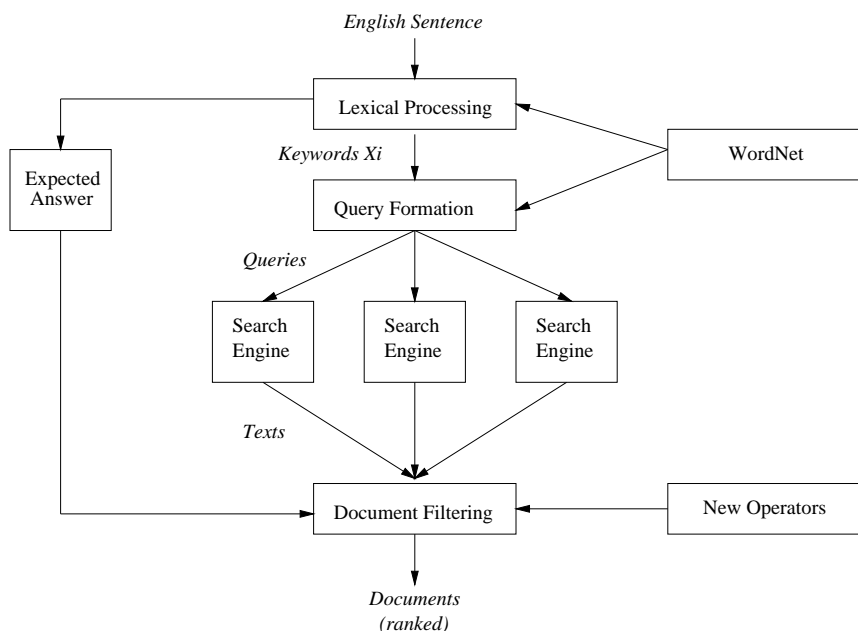


Figure 2: System organization

3.1 Lexical processing

This module has been adopted from an information extraction system developed by us for the MUC competition [Moldovan et al. 1993]. First the sentence boundaries are located. It then places part of speech tags on words by using a version of Brill’s tagger [Brill 1992] in conjunction with WordNet. It also contains a phrase parser that segments each sentence into constituent noun and verb phrases and recognizes the head words. After the elimination of the stopwords (conjunctions, prepositions, pronouns and modal verbs) we are left with some

keywords x_i that represent the important concepts of the input sentence.

Word-sense disambiguation

Our idea for word sense disambiguation is to use WordNet to determine the possible association between words. The approach we are using for this task, described in [Mihalcea and Moldovan 1998], is based on the idea of *semantic density*. This can be measured by the number of common words that are within a semantic distance of two or more words. The closer the semantic relationship between two words the higher the semantic density between them. We introduce the semantic density because it is relatively easy to measure it on a machine readable dictionary like WordNet. This is done by counting the number of concepts two words have in common. A metric is used in this sense, which when applied to all possible combinations of the senses of two or more words it ranks them.

Given a pair of words $X - Y$, two successive algorithms are applied to identify the most likely combinations of their senses. First of all, the senses of Y are ranked, using the Internet for gathering statistics; then the semantic distance between the words is measured using WordNet.

The first step of this method creates sets of pairs in which one of the words remains constant, i.e. X , and Y is replaced by the words in its similarity lists. Using WordNet, a similarity list is created for each sense of Y , and it contains the words from Y 's synset. A search performed on the Internet for each of these sets of pairs will provide a number of hits, and thus will indicate a ranking over the possible senses of the word Y .

In WordNet each concept has a gloss, an explanation in English of the meaning of that concept. The conceptual distance between two words X and Y is determined by counting the number of common concepts between the words semantically related to X (i.e. from the X 's WordNet hierarchy) and the words that can occur in the context of Y (i.e. the words from Y 's gloss).

Consider for example the disambiguation of the words *revise* and *law* in the phrase “*revise law*”. In WordNet 1.5, the verb *revise* has 2 possible senses and the noun *law* has 7 possible senses. We searched on Internet, using AltaVista, for all possible pairs verb-noun that may be created using *revise* and the words from the similarity lists of *law*. Over the seven possible senses for this noun, the first step of our method indicated the following ranking (we indicate the number of hits between parenthesis): *law*#2(2829), *law*#3(648), *law*#4(640), *law*#6(397), *law*#1(224), *law*#5(37), *law*#7(0). Thus, only the sense #2 and #3 of the noun *law* are eligible to be used for the next algorithm.

For each of the two senses of the verb, we determined the noun-context sv_k ; the noun-context includes the nouns from the glosses in the sub-hierarchy of the verb, and the associated weights w_k (the weights are given by the level within the verb hierarchy). For each of the two possible senses of the noun, we determined the list of all the nouns from the noun's hierarchy sn_l .

The common words between these two lists sv_k and sn_l will produce a list of common concepts with the associated weights w_k . The conceptual density between the noun and the verb is given by the formula:

$$C_{ij} = \frac{\sum_k^{|cd_{ij}|} w_k}{\log(desc_i)}$$

where: $|cd_{ij}|$ is the number of common concepts; w_k are the weights associated with the nouns

from the noun-context of the verb; $desc_i$ is the total number of words within the hierarchy of the noun.

In Table 2, we present:

- a) the values obtained for the combinations of different senses, i.e. the number of common concepts between the verb and noun hierarchies - $|cd_{ij}|$ (columns 2-3)
- b) the summations of the weights associated with each noun within the noun-context of the verb v_j (columns 4-5)
- c) the total number of nouns within the hierarchy of each sense n_i i.e. $desc_i$ (columns 6-7)
- d) the conceptual density C_{ij} for each pair $n_i - v_j$, derived using the formula presented above (columns 8-9)

	$ cd_{ij} $		weights		$desc_i$		C_{ij}	
	2	3	4	5	6	7	8	9
	n_2	n_3	n_2	n_3	n_2	n_3	n_2	n_3
v_1	5	4	2.06	2	975	1265	0.30	0.28
v_2	0	0	0	0	975	1265	0	0

Table 2: Values used in computing the conceptual density; v_i indicates the sense number i of the verb and n_i indicates the sense number i of the noun

The biggest value for conceptual density is given by $v_1 - n_2$:

$$revise\#1/2 - law\#2/5 \quad C_{11} = 0.30$$

This combination of verb-noun senses³ appears in SemCor, file br-a01.

An essential aspect of the word sense disambiguation method used here is that we provide a ranking of possible associations between words instead of a binary yes/no decision for each possible sense combination. This allows for a controllable precision even in the case of short sentences of questions. Usually the questions or sentences used for search on the Internet are ambiguous by having a poor context, still the results of a search or interface based on such a sentence can be improved if the possible associations between the senses of the verb and the noun are determined.

We have tested our disambiguation method against the semantic annotations from SemCor [Miller, Leacock et al., 1993], and the results showed that in 80% of the cases the correct result as indicated by SemCor was in the top four choices of the ranked list of possible pairs of the two words senses.

3.2 Query formulation

The two main functions performed by this module are: 1) the construction of similarity lists using WordNet, and 2) the actual query formation.

Once we have a sense ranking of each word in the input sentence, it is relatively easy to use the rich semantic information contained in WordNet to identify many other words that are semantically similar to a given input word. By doing this we increase the chance of finding more answers to input queries. WordNet can provide semantic similarity between concepts at various levels. Here are three levels that may be considered in descending order of interest.

Level 1. Words are semantically similar if they belong to the same synset.

³The notation $\#i/n$ means sense i out of n possible.

Level 2. Words that express concepts linked by semantic relations; i.e. hyponymy/ hypernymy, meronymy/ holonymy and entailment.

Level 3. Words that belong to sibling concepts; namely concepts that are subsumed by the same concept.

Consider, for example, the sense number 1 of the word *activity*. There are 7 senses distinguished by WordNet for this word. The synset for the first sense includes two other synonyms *action* and *activeness*. The hypernym synset for this first sense includes only one word: *state*. The similarity list that we can now create for this word are:

$W = \{ \textit{action}, \textit{activity}, \textit{activeness} \}$ considering the first level of similarity, respectively

$W' = \{ \textit{action}, \textit{activity}, \textit{activeness}, \textit{state} \}$ for the second similarity level.

Several experiments have been performed in order to measure the performance achieved using different levels of word similarities. Conclusions drawn from experiments on small collections of texts [Salton and Lesk 1971] showed that expansion by synonyms improved the performance, but expansion by broader or narrower terms, selected from a hierarchical thesaurus did not prove to be very useful.

[Voorhees 1994] investigated the efficacy of expanding a query for search in large text collections. The work developed during this investigation uses WordNet and experiments for four expanding strategies: expansion by synonyms only, expansion by synonyms plus all descendents in a *isa* hierarchy, expansion by synonyms plus parents and all descendents in a *isa* hierarchy, and expansion by synonyms plus any synset directly related to the given synset. The results have shown that there are no significant differences between the precision obtained while using the four expanding strategies.

These experiments performed previously by researchers, and several other tests that we performed, drove us to the conclusion that no important improvement can be achieved by using broader or narrower terms. Thus, the results that we are presenting include only the *Level 1* similarity.

Let's denote with x_i the words of a question or sentence, and with $W_i = \{x_i, x_i^k\}$ the similarity lists provided by WordNet for each word x_i . The elements of a list are x_i^k where k enumerates the elements in each list, i.e. words on the same level of similarity with the word x_i . These lists can now be used for the actual query formulation, using the boolean operators accepted by the current search engines. The *OR* operator is used to link words within a similarity list W_i , while the *AND* and *NEAR* operators link the similarity lists.

While different combinations of similarity lists linked by *AND* or *NEAR* operators are possible, there are two basic forms giving the maximum, respectively the minimum, of the number of documents retrieved:

(1) $W_1 \text{ AND } W_2 \text{ AND } \dots \text{ AND } W_n$

(2) $W_1 \text{ NEAR } W_2 \text{ NEAR } \dots \text{ NEAR } W_n$

In most of the cases, the format (1) gathered thousands of documents, while the format (2) had almost always null results.

The conclusion so far is that the documents containing the answers, if any, must be among the large number of documents provided by the AND operators. However, the search engines failed to rank them in the top of the list. Thus, we sought to find new operators that filtered out many of the irrelevant texts.

3.3 New operators

Our approach to filtering documents is to first search the Internet using weak operators (AND, OR) and then to further search this large number of documents using more powerful operators. For this second phase, we propose the following additional operators:

PARAGRAPH n (... similarity lists ...)

The PARAGRAPH operator searches like an AND operator for the words in the similarity lists with the constraint that the words belong only to some n paragraphs. The rationale is that most likely the information requested is found in a few paragraphs rather than being dispersed over an entire document. A similar idea can be found in [Callan 1994].

SENTENCE n (... similarity lists ...)

The SENTENCE operator searches like an AND operator for the words in the similarity lists with the constraint that the words belong to a sentence. The answers to many queries are found in single, sometimes complex sentences.

SEQUENCE ($W_1 d W_2 d \dots W_n$)

where d is a numeric variable that indicates the distance between the words in the W lists for which the search is done. The SEQUENCE operator imposes a more flexible NEAR search, but it requires that the sequence of the words in the similarity list be maintained as specified. Of course, combinations of these operators are possible.

From these three new operators, we practically implemented the first two. For this, the documents gathered from the Internet have to be segmented into sentences and paragraphs, respectively. Separating a text into sentences proves to be an easy task, one could just make use of the punctuation to solve this problem. Instead, paragraph segmentation is much more difficult, and this is due first of all to the highly unstructured texts that can be found on the Web. Work developed in this direction is presented in [Hearst 1994] and [Callan 1994]. But these methods work only for structured texts, containing a priori known lexical separators (i.e. a tag, an empty line e tc.). Thus, we had to use a method that covers almost all the possible paragraph separators that can occur in the texts on the web. The paragraph separators that we considered so far are: (1) HTML tags; (2) empty lines; (3) paragraph indentation.

4 An example

The system operation is presented below with the help of an example. Suppose one wants to find the answer to the question: ‘‘How much tax an average salary person pays in the United States?’’

The linguistic processing module identified the following keywords:

$x_1 = (\text{tax}), \text{pos} = \text{noun}, \text{sense} \#1/1$

$x_2 = (\text{average}), \text{pos} = \text{adjective}, \text{sense} \#4/5$

$x_3 = (\text{salary}), \text{pos} = \text{noun}, \text{sense} \#1/1$

$x_4 = (\text{the United States}), \text{pos} = \text{noun}, \text{sense} \#1/2$

$x_5 = (\text{person}), \text{pos} = \text{noun}, \text{sense} \#1/3$

$x_6 = (\text{pays}), \text{pos} = \text{verb}, \text{sense} \#1/7$

In the notation above ‘‘pos’’ means part of speech, and the sense number indicates the actual WordNet sense that resulted from the disambiguation out of all possible senses in WordNet.

For instance adjective **average** has 5 senses and the system picked sense #4. Note that the senses of words in WordNet are ranked in the order of their utilization frequency in a large corpora.

These keywords will be the input for the next step of our system, except those keywords having a too high position in the WordNet hierarchies. In our example considering only *Level 1* similarity, and only the first four keywords WordNet provides:

$W_1 = \{\text{tax, taxation, revenue enhancement}\}$

$W_2 = \{\text{average, intermediate, medium, middle}\}$

$W_3 = \{\text{salary, wage, pay, earnings, remuneration}\}$

$W_4 = \{\text{United States, United States of America, America, US, U.S., USA, U.S.A.}\}$

These lists are used to formulate queries for the search engine. As we will see, the operators available today for the search engines are not adequate to provide the desired answers in most of the cases. Table 3 shows some queries and the number of documents provided by AltaVista, considered to be one of the search engines with the most powerful set of operators available today.

	Query	Number of documents
1	$W_1 \text{ AND } W_2 \text{ AND } W_3 \text{ AND } W_4$	15,464
2	$W_1 \text{ AND } (W_2 \text{ NEAR } W_3) \text{ AND } W_4$	3,217
3	$W_1 \text{ NEAR } (W_2 \text{ NEAR } W_3) \text{ AND } W_4$	803
4	$W_1 \text{ NEAR } W_2 \text{ NEAR } W_3 \text{ NEAR } W_4$	0
5	$W_1 \text{ AND } \{\text{average } W_3\} \text{ AND } W_4$	1752
6	$W_1 \text{ AND } \{\text{average } W_3\} \text{ NEAR } W_4$	1 (no)
7	$W_1 \text{ NEAR } \{\text{average } W_3\} \text{ NEAR } W_4$	0

Table 3: Queries with various combinations of operators

The ranking provided by the Alta Vista is of no use for us here. None of the leading documents in any category provides the desired information. The only document fetched by Query 6 is equally irrelevant:

....Instead, their plans would shift more of the total tax burden on to labor, taxing capital once under a business tax, and taxing wages and salaries twice under both the income tax and the payroll tax. Middle-class Americans have to pay more under such a system, and wealthy people much less....

....The average taxpayer must work 86 days to pay all federal taxes, and must work 36 days just to pay his or her federal income tax. The average American must work 2 hours and 49 minutes every working day to pay all their taxes.....

An analysis of the results in table above indicates that there is a gap in the volume of documents retrieved with the Alta Vista operators. For instance using only the AND operator (Query 1) 15,464 documents were obtained, but the NEAR operator (Query 4) produced no output. This operator seems to be too restrictive, while it fails to identify the right answer. Various combinations of AND and NEAR operators were tried, as indicated by the table above with no great results.

Using the PARAGRAPH operator for the example above, the system found a relevant answer:

In 1910, American workers paid no income tax. In 1995, a worker earning an average wage of \$26,000 pays about 24% (about \$6,000) in income taxes. The average American worker's pay has risen greatly since 1910. Then, the average worker earned about \$600 per year. Today, the figure is \$26,000.

5 Evaluation of Results

The system has been tested on 50 questions derived from the TREC-6 corpus. Table 4 presents ten randomly selected questions from this set, together with the results we obtained.

Question	AND	NEAR	AND	NEAR	Paragraph	Sentence
	x_i	x_i	w_i	w_i	w_i	w_i
Which are some of the organizations participating in international criminal activity?	27,716 0	3 1	48,133 0	5 1	6 1	0 0
Is the disease of Poliomyelitis (polio) under control in the world?	9,432 1	13 3	10,271 2	15 3	40 11	3 2
Which are some of the positive accomplishments of the Hubble telescope since it was launched?	178 1	4 0	504 1	4 0	2 1	0 0
Which are some of the endangered mammals?	32,133 0	6,214 1	32,133 0	6,214 1	150 80	5 2
Which are the most crashworthy, and least crasworthy, passenger vehicles?	246 0	5 1	260 1	5 1	15 6	1 0
How many civilian non-combatants have been killed in the various civil wars in Africa?	188 0	0 0	283 0	0 0	11 3	0 0
What are the advantages and/or disadvantages of tooth implants?	1,722 0	17 2	3,935 0	39 3	0 0	0 0
Is there some evidence that radio waves from radio towers or car phones affect brain cancer occurrence?	1,392 0	0 0	5,684 0	0 0	9 6	2 0
Which is the commercial use of magnetic levitation?	696 1	0 0	909 2	0 0	38 21	3 1
Which are the roots and prevalence of polygamy in the world today?	180 0	0 0	459 1	0 0	1 1	0 0

Table 4: A sample of the results obtained for randomly selected questions from the TREC collection. In each box, the top number indicates the number of documents retrieved, and the bottom number indicates the relevant documents in top 10 ranking, and respectively the total number of relevant paragraphs.

Each cell in this table includes two numbers: the upper one represents the total number of documents retrieved for the question, respectively the total number of paragraphs retrieved when the PARAGRAPH operator was used. The bottom number represents the number of relevant documents found in top 10 ranking, respectively the number of relevant paragraphs.

The AND x_i and NEAR x_i columns contain the results for the search when AND and NEAR operators were applied to the input words x_i . By replacing the words x_i with their similarity lists derived from WordNet, the number of documents retrieved was increased, as expected. The results obtained in these cases, with an AND, respectively a NEAR operator applied to the similarity lists, are presented in the columns AND w_i and NEAR w_i .

The next column contains the number of documents extracted when the new operator PARAGRAPH 2 (meaning two consecutive paragraphs) was applied to words from the similarity lists. The results were encouraging. The number of documents retrieved was small and correct answers were found in all cases.

The last column shows that the operator SENTENCE was too restrictive, producing correct answers in only three out of ten cases.

A summary of the results, for the 50 questions that we used to test our system, is presented in Table 5. It is interesting to observe that the query extension determined an increase in the number of documents, by a number of times varying from 0 (meaning equal number of documents retrieved for both the unextended and extended queries) to 32.

Question	AND x_i	NEAR x_i	AND w_i	NEAR w_i	PARAGRAPH w_i
Average question	7,746	258	25,803	332	26.04
	0.16	0.48	0.44	0.88	11.10

Table 5: Summary of results for the 50 questions from TREC collection. In each box, the top number indicates the number of documents retrieved, and the bottom number indicates the relevant documents in top 10 ranking, and respectively the total number of relevant paragraphs.

The number of relevant documents found in top 10 ranking was also increased for the search performed with an extended query. As it can be seen from the summary results presented in the table above, the query extension increased the number of documents retrieved, and also the number of relevant documents in top 10 ranking.

With the PARAGRAPH operator, the number of relevant documents (paragraphs) retrieved decreased significantly. For the 50 tests that we performed, we obtained an overall *precision* of 43% while using this operator.

It is hard to compare our system performance with the performance achieved by other implementations; systems implemented so far, that try to retrieve answers for narrow questions, (1) address specific searches, as for example [FindLaw] which is designed for finding legal resources on Internet; (2) address narrower domains on the Web, as the system described in [Burke, Hammond et al. 1995] which uses the files of “Frequently Asked Questions” (FAQs) associated with many Usenet groups; (3) are designed to retrieve information not on Internet, but on very large collections of texts [Voorhees 1994].

In [Voorhees 1994] it is reported an average precision of 36% for full topic statements. Our result of 43% *precision*, in retrieving information for narrow questions on heterogeneous domains on Internet, is thus encouraging.

6 Conclusions

This paper has introduced the idea of using WordNet to extend the Web search based on semantic similarity. The example clearly shows that without this it was not possible to find an answer. Then, we have introduced some new operators that fill the gap between the operators currently used by the search engines.

The broad use of natural language queries in information retrieval is still beyond the capabilities of current natural language technology. Machine readable dictionaries, such as WordNet, prove to be useful tools to web search. However, their use for the Internet has been limited so far [Allen 1997], [Hearst, Karger et al. 1995], [Katz 1997].

There are several other possible ways of improving the web search not discussed in this paper. One such a possibility is to index words by their WordNet senses. This of course

implies some on-line word-sense disambiguation of documents which may be possible in not too distant future. Semantic indexing has the potential of improving the ranking of search results, as well as to allow information extraction of objects and their relationships [Pustejovsky, Boguraev et al. 1997].

Another way to improve the web search is to use compound nouns or collocations. In WordNet there are thousands of groups of words such as *blue color worker*, *stock market*, etc., that point to their respective concept. Each compound noun is better indexed as one term. This reduces the storage space for the search engine and may increase the precision.

References

- [AltaVista] Digital Equipment Corporation. AltaVista Home Page.
<http://www.altavista.digital.com>.
- [Allen 1997] Allen, B.P. WordWeb - Using the Lexicon for WWW. *Inference Corporation*
<http://www.inference.com> 1997
- [Anikina, Golender et al. 1997] Anikina, N.; Golender, V.; Kozhukhina, S.; Vainer, L. and Zagatsky, B. REASON: NLP-based Search System for WWW. Proceedings of the American Association for Artificial Intelligence Conference, Spring Symposium, "NLP for WWW", 1-10, Stanford University, CA, 1997.
- [Brill 1992] Brill, E. A simple rule-based part of speech tagger. Proceedings of the 3rd Conference on Applied Natural Language Processing, Trento, Italy, 1992
- [Burke, Hammond et al. 1995] Burke, R.; Hammond, K. and Kozlovsky J. Knowledge-based Information Retrieval from Semi-Structured Text. Proceedings of the American Association for Artificial Intelligence Conference, Fall Symposium, "AI Applications in Knowledge Navigation & Retrieval", 15-19, Cambridge, MA, 1995.
- [Callan, Croft et al. 1992] Callan J.P.; Croft W.B. and Harding S.M. The INQUERY Retrieval System. Proceedings of the 3rd International Conference on Database and Expert Systems Applications, 78-83, 1992.
- [Callan 1994] Callan, J.P. Passage-Level Evidence in Document Retrieval. Proceedings of the 17th Annual International ACM SIGIR, Conference on Research and Development in Information Retrieval, 302-310, Dublin, Ireland, 1994.
- [FindLaw] FindLaw, Internet Legal Resources <http://www.findlaw.com/index.html>
- [Gravano, Chang et al. 1997] Gravano, L.; Chang, K.; Garcia-Molina, H.; Lagoze, C. and Paepcke, A. STARTS Stanford Protocol Proposal for Internet Retrieval and Search. Digital Library Project, Stanford University, 1997.
- [Hearst 1994] Hearst, M.A. Multi-paragraph segmentation of expository text. Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics, 9-16, Las Cruces, New Mexico, 1994.
- [Hearst, Karger et al. 1995] Hearst, M.A.; Karger D.R. and Pedersen, J.O. Scatter/Gather as a Tool for the Navigation of Retrieval Results. Proceedings of the American Association for Artificial Intelligence Conference, Fall Symposium "AI Applications in Knowledge Navigation & Retrieval", 65-71, Cambridge, MA, 1995

- [Katz 1997] Katz, B. From Sentence Processing to Information Access on the World Wide Web, Proceedings of the American Association for Artificial Intelligence Conference, Spring Symposium, "NLP for WWW", 77-86, Stanford University, CA, 1997
- [Mihalcea and Moldovan 1998] Mihalcea, R. and Moldovan, D.I. Word Sense Disambiguation Based on Semantic Density. To appear in Proceedings of COLING-ACL '98 Workshop on Usage of WordNet in Natural Language Processing Systems, Montreal, Canada, 1998.
- [Miller, Leacock et al., 1993] G.A. Miller, C. Leacock, T. Randee and R. Bunker, A Semantic Concordance. Proceedings of the 3rd DARPA Workshop on Human Language Technology, 303-308, Plainsboro, New Jersey, 1993
- [Miller 1995] Miller, G.A. WordNet: A Lexical Database. Communication of the ACM, 38(11):39-41.
- [Moldovan et al. 1993] Moldovan, D. et al. USC: Description of the SNAP System Used for MUC-5. Proceedings of the 5th Message Understanding Conference, Baltimore, MD, 1993
- [Pustejovsky, Boguraev et al. 1997] Pustejovsky, J.; Boguraev B., Verhagen, M.; Buitelaar, P. and Johnston, M. Semantic Indexing and Typed Hyperlinking. Proceedings of the American Association for Artificial Intelligence Conference, Spring Symposium, "NLP for WWW", 120-128. Stanford University, CA, 1997.
- [Salton and Lesk 1971] Salton, G. and Lesk, M.E. Computer evaluation of indexing and text processing. Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, 143-180, Prentice Hall, Englewood Cliffs, New Jersey 1971.
- [Selberg and Etzioni 1995] Selberg, E. and Etzioni, O. Multi-Service Search and Comparison Using the MetaCrawler. Proceedings of the 4th International World Wide Web Conference, 195-208, Boston, MA.
- [TREC 1997] Text REtrieval Conference
<http://trec.nist.gov> 1997
- [Voorhees 1994] Voorhees, E.M. Query Expansion using Lexical-Semantic Relations Proceedings of the 17th Annual International ACM SIGIR, Conference on Research and Development in Information Retrieval, 61-69, Dublin, Ireland, 1994.
- [Zorn, Emanoil et al. 1996] Zorn, P.; Emanoil, M. and Marshall, L. Advanced Searching: Tricks of the Trade. *Online. The Magazine of Online Information Systems* 20(3), 1996