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

July 21, 1999

Not for distribution or attribution. For review purposes only.

Abstract

This paper presents a natural language interface system to an Internet search engine that provides the following improvements: (1) accepts natural language (English) questions, (2) expands the query, based on a word sense disambiguation method, and (3) uses a new lexical operator to post-process the documents retrieved for extracting only the part of a document that is relevant to a query. The system was tested on 100 queries of which 50 were adopted from the TIPSTER topics collection, provided at the 6th Text Retrieval Conference (TREC-6) and 50 were selected from among the queries submitted by users to an existing Web search engine. The results obtained demonstrate a substantial increase in both the precision and the percentage of queries answered correctly, while the amount of text presented to the user is reduced in comparison with the current Internet search engine technology.

1 Introduction

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, and (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 existing search engines to improve the quality of the information retrieved.

A main problem with the current search engines is that broad, general queries produce a large volume of documents extracted, while specific, narrow questions often fail to produce any documents [Selberg and Etzioni 1995], [Zorn, Emanoil et al. 1996]. Many of the documents retrieved for general queries are totally irrelevant and many relevant documents are missing because the query does not contain the keywords that index those documents. Some queries formulated in terms of restrictive boolean operators lead to the right documents, but most often being too restrictive, these queries extract no documents.

Another problem that hinders the use of the search engines is the lack of a natural language interface. Many users who are non-computer professionals would prefer to use natural language questions instead of the rather complex boolean expressions currently accepted by the search engines. For example, for finding the presidents of the United States during the last century, a common user would ask *"Who were the US Presidents of the last century?"*, instead of forming a query with boolean operators, such as *(US NEAR Presidents) AND (last NEAR century)*.

Undoubtedly, a natural language interface capable of transforming sentences into queries with boolean operators currently accepted by the search engines would be beneficial. But there is yet another, perhaps even greater advantage in using English questions. With a modest amount of linguistic processing it is possible to disambiguate the words of a query and then search not only for the words of the input sentence, but create *similarity lists* with words from on-line dictionaries that have the same meaning as the input words. Thus the original English query is expanded automatically into many boolean queries which can significantly broaden the web search.

The other area of improvement is to design better information retrieval operators that further filter the documents returned by an Internet search engine. Instead of returning to the user many, often large documents that are impractical to inspect, it is possible to return only relevant paragraphs.

In this paper we describe a system that performs query expansion using WordNet, an on-line lexical database developed at Princeton University [Miller 1995]. Unlike other information retrieval systems that perform query expansion by simply using word synonyms, our system first disambiguates the senses of the query words and then searches only for the words that are synonyms with the query semantic concepts. The large number of documents that result from the Internet search are then subject to a new search using a new operator called PARAGRAPH. This operator extracts only the paragraphs that render relevant information to a query. The goal in a question answering system like ours is not to retrieve entire documents as normally done in information retrieval, but to provide the user with answers. Since our system does not have a module to formulate answers, it simply returns to the user the paragraphs that may contain the answer.

The performance of information retrieval systems that operate on known sets of input documents is measured based on 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 the ratio between the number of relevant documents retrieved over the total number of documents retrieved, and the *recall* is the ratio between the number of relevant documents extracted over the total number of relevant documents in the database. Unfortunately, when searching the Internet, the difficulty is that the number of input documents and the number of relevant documents are unknown. For us, the ultimate performance test is whether or not the system provides a correct answer. Thus, we introduce in this paper a new performance measure, the system *productivity*, which is the percentage of questions answered satisfactorily.

2 Related work

Several approaches have been previously considered to improve the quality of the Internet search and the performance of information retrieval systems. One idea is to use 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 difficulty 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.

Natural Language Processing techniques have been used for information retrieval. Here, work has been developed in two directions: (1) the use of query extension techniques to increase the number of documents retrieved, and (2) the improvement of the quality of the information retrieved: REASON [Anikina, Golender et al. 1997], INQUIRY [Callan, Croft et al. 1992].

Query expansion has long been used to retrieve more relevant information [Salton and Lesk 1971]. 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; [Voorhees 1994] and [Allen 1997] used WordNet to enlarge the query such as it includes words which are semantically related to the concepts from the original query. The basic semantic relation used in their systems is the *synonymy* relation; still, these techniques allow a further extension of the query, by using other semantic relations which can be derived from a MRD, like for example the hypernymy and hyponymy relations.

A second technique considered by researchers for query expansion is to use words derived from relevant documents. The SMART system, [Buckley et al. 1994], developed at Cornell University, does massive query expansion, adding from 300 to 530 terms to each query, terms which are acquired from relevant documents. They report a precision improvement of 7% to 25% obtained during their experiments. Another method proposed by [Ishikawa, Satoh and Okumura 1997] extends the original query with words from paragraphs which are considered to be relevant, based on a similarity measure between the paragraphs and the original query. [Lu and Keeffer 1994] evaluated the performance obtained with query extension techniques during the experiments performed with the TIPSTER collection; they observed that larger queries can increase the precision within a range from 0% to 20%.

3 Background on resources

Several 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 extracting information makes use of the AltaVista search engine. The system has been tested on 100 questions of which 50 questions were derived from the topics provided at the 6th Text Retrieval Conference (TREC-6).

3.1 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 160,000,000 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 the documents containing all of the specified words or phrases, OR finds the documents containing at least one of the specified words or phrases, and $NEAR$ finds the documents containing all specified words or phrases that are within 10 words of each other.[1]

Our main concern when we decided to rely on AltaVista for searching documents on Internet, regarded the reliability of this search engine. The number of hits obtained for a query should vary only within a small range, for searches performed at different time intervals. To test the reliability of AltaVista, we considered a set of 1,100 words (nouns, verbs, adjectives and adverbs); the set was built from one of the texts in the Brown corpus. A test run consisted of searching the Internet using AltaVista, for each of these words, and recording the number of hits obtained. Twenty tests were performed using the same words, over o period of 10 days, a test run at every 12 hours. The overall results for these tests showed that, given an average of the number of hits AV for a particular word:

- 90% of the time the hits are in the range [0.99AV - 1.01AV]
- 100% of the time the hits are in the range [0.85AV - 1.15AV]

Taking into consideration the size of the information found on the Internet and the fact that this information is highly unstructured, the small variations achieved by AltaVista in searching the Internet can classify this search engine as a reliable one.

3.2 WordNet

WordNet[2] is a Machine Readable Dictionary developed at Princeton University by a group led by George Miller [Miller 1995], [Fellbaum 1998]. 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 sysnset represents a concept. WordNet has a large network of 129,509 words, organized in 99,643 synonym sets, called *synsets.* There is a rich set of 299,711 relation links among words, between words and synsets, and between synsets.

As an example of a word representation in WordNet, let us consider the noun *computer.* It has two senses defined in WordNet, hence it belongs to two synsets: {*computer, data processor, electronic computer, information processing system*} and {*calculator, reckoner, figurer, estimator, computer*}.

[1] These examples are from the *AltaVista Advanced Help* $http://www.altavista.digital.com/av/content/help\_advanced.htm$

[2] WordNet 1.6 has been used in our algorithm implementation.

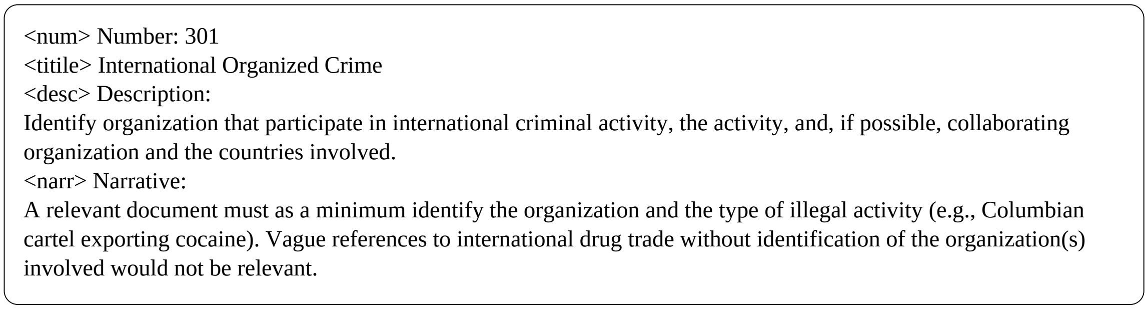
3.3 TREC topics

The Text Retrieval Conferences (TREC) are part of the TIPSTER Program, and are intended to encourage research in information retrieval from a large number of 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. Fifty natural language questions were derived 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-liked data structure. Its fields are as follows: the <*num*> section identifies the topic number; 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
<titile> 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 derived was: `‘‘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.

4 System architecture

The system architecture is shown in Figure 2. The input query or sentence expressed in English is first presented to the lexical processing module. This module was adopted from an information extraction system developed by us for the MUC competi-

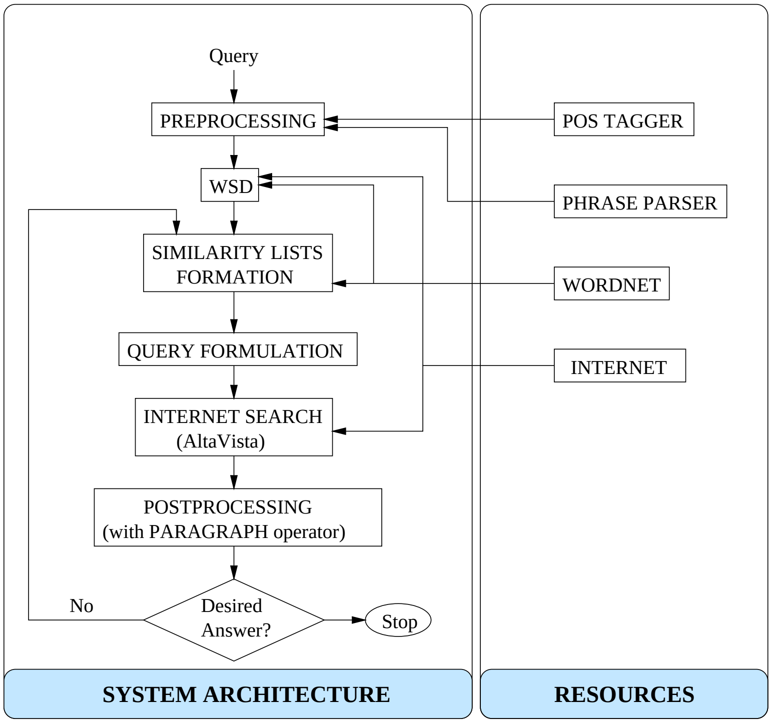


Figure 2: System Architecture

tion [Moldovan et al. 1993]. The word and sentence boundaries are located via a process called tokanization. The words are part-of-speech tagged by using a version of Brill's tagger [Brill 1992]. A phrase parser 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. The rest of the modules are described below.

5 Word-sense disambiguation

A novelty of our system as compared to other information retrieval systems is that it performs the sense disambiguation of the keywords. This means that each keyword in the query is mapped into its corresponding semantic form as defined in WordNet. This step is necessary in order to be able to expand the queries based on their semantic concepts and not based on the keywords.

The approach we use for the word sense disambiguation takes advantage of the sentence context. The words are paired and each word is disambiguated in the context of the other word. This is done by searching on the Internet with queries formed using different senses of one word while keeping the other word fixed. The senses are ranked simply in the order provided by the number of hits. In this way all the words are processed and senses are ranked. The next step is to refine the ordering of senses by using a completely different method, namely the *semantic density*. This is measured by the number of common words that are within a semantic distance of two or more words and it is done using the WordNet glosses. The algorithms and the performance results are presented below.

5.1 Contextual ranking of word senses

5.1.1 Algorithm 1

Input: semantically untagged word$_1$ - word$_2$ pair ($W_1 - W_2$)
Output: ranking the senses of one word
Procedure:

1. *Form a similarity list for each sense of one of the words.*
   Pick one of the words, say W_2, and using WordNet, form a similarity list for each sense of that word. For this, use the words from the synset of each sense of the word. Consider, for example, that W_2 has m senses. This means that W_2 appears in m similarity lists:
   $(W_2^1, W_2^{1(1)}, W_2^{1(2)}, ..., W_2^{1(k_1)})$
   $(W_2^2, W_2^{2(1)}, W_2^{2(2)}, ..., W_2^{2(k_2)})$
   ...
   $(W_2^m, W_2^{m(1)}, W_2^{m(2)}, ..., W_2^{m(k_m)})$
   where W_2^1, W_2^2, ..., W_2^m are the senses of W_2, and $W_2^{i(s)}$ represents the synonym number s of the sense W_2^i as defined in WordNet.

2. *Form* $W_1 - W_2^{i(s)}$ *pairs.* The pairs that may be formed are:
   $(W_1 - W_2^1, W_1 - W_2^{1(1)}, W_1 - W_2^{1(2)}, ..., W_1 - W_2^{1(k_1)})$
   $(W_1 - W_2^2, W_1 - W_2^{2(1)}, W_1 - W_2^{2(2)}, ..., W_1 - W_2^{2(k_2)})$
   ...
   $(W_1 - W_2^m, W_1 - W_2^{m(1)}, W_1 - W_2^{m(2)}, ..., W_1 - W_2^{m(k_m)})$

3. *Search the Internet and rank the senses* $W_2^{i(s)}$.
   A search performed on the Internet for each set of pairs as defined above, results in the number of hits indicating the frequency of occurrences for W_1 together with that sense of W_2.

   Using the operators provided by AltaVista, form a query for each set above. One such query is:

   ("W_1* W_2^i*" OR "W_1* $W_2^{i(1)}$*" OR "W_1* $W_2^{i(2)}$*" OR ... OR "W_1* $W_2^{i(k_i)}$*") for all $1 \leq i \leq m$. The asterisk (*) is used as a wildcard to increase the number of hits with morphologically related words. Using such a query, we get the number of hits for each sense i of W_2 and this provides a ranking of the m senses of W_2 as they relate with W_1.

A similar algorithm is used to rank the senses of W_1 while keeping W_2 constant (undisambiguated). Since these two procedures are done over a large corpora (the Internet), and with the help of similarity lists, there is little correlation between the results produced by the two procedures.

Procedure Evaluation This method was tested on 384 pairs: 200 verb-noun, 127 adjective-noun, and 57 adverb-verb extracted from the first text of the SemCor 1.6 from the Brown corpus. Using the query form presented above on Alta Vista, we obtained the results shown in Table 1. The table indicates the percentages of correct senses (as given by SemCor) ranked by us in top 1, top 2, top 3, and top 4 of our list. We concluded that by keeping the top four choices for verbs and nouns and the top two choices for adjectives and adverbs, we cover with high percentage (mid and upper 90's) all relevant senses. Looking from a different point of view, the possible use of the procedure so far is *to exclude* the senses that do not apply, and this can save considerable amount of computation time as many words are highly polysemous.

	top 1	top 2	top 3	top 4
noun	76%	83%	86%	98%
verb	60%	68%	86%	87%
adjective	79.8%	93%		
adverb	87%	97%		

Table 1: Statistics gather from the Internet for 384 word pairs.

5.2 Further ranking of senses using the conceptual density

A measure of the relatedness between words can be a knowledge source for several decisions in NLP applications. The approach taken here is to construct a linguistic context for each sense of the verb and noun, and to measure the number of the common nouns shared by the verb and the noun contexts. In WordNet each concept has a gloss that acts as a micro-context for that concept. This is a rich source of linguistic information that we found useful to determine conceptual density between words. This method is applicable only to the verb - noun pairs, the adjectives and adverbs cannot benefit from this algorithm.

5.2.1 Algorithm 2

Input: semantically untagged verb - noun pair and a ranking of noun senses (as determined by Algorithm 1)
Output: sense tagged verb - noun pair
Procedure:

1. Given a verb-noun pair $V - N$, denote with $< v_1, v_2, ..., v_h >$ and $< n_1, n_2, ..., n_l >$ the possible senses of the verb and the noun using WordNet.

2. Using Algorithm 1, the senses of the noun are ranked. Only the first t possible senses of this ranking will be considered. The rest are dropped to reduce the computational complexity.

3. For each possible pair $v_i - n_j$, the conceptual density is computed as follows:

   (a) Extract all the glosses from the sub-hierarchy including v_i (the rationale for selecting the sub-hierarchy is explained below).

   (b) Determine the nouns from these glosses. These constitute the noun-context of the verb. Each such noun is stored together with a weight w that indicates the level in the sub-hierarchy of the verb concept in whose gloss the noun was found.

   (c) Determine the nouns from the noun sub-hierarchy including n_j.

   (d) Determine the conceptual density C_{ij} of common concepts between the nouns obtained at (b) and the nouns obtained at (c) using the metric:

$$C_{ij} = \frac{\sum_{k}^{|cd_{ij}|} w_k}{log(descendents_j)} \qquad (1)$$

   where:

- $|cd_{ij}|$ is the number of common concepts between the hierarchies of v_i and n_j
- w_k are the levels of the nouns in the hierarchy of verb v_i, and
- $descendents_j$ is the total number of words within the hierarchy of noun n_j

4. C_{ij} ranks each pair $v_i - n_j$, for all i and j.

Rationale

1. In WordNet, a gloss explains a concept and provides one or more examples with typical usage of that concept. In order to determine the most appropriate noun and verb hierarchies, we performed some experiments using SemCor and concluded that the noun sub-hierarchy should include all the nouns in the class of n_j. The sub-hierarchy of verb v_i is taken as the hierarchy of the highest hypernym h_i of the verb v_i. It is necessary to consider a larger hierarchy then just the one provided by synonyms and direct hyponyms. As we replaced the role of a corpora with glosses, better results are achieved if more glosses are considered. Still, we do not want to enlarge the context too much.

2. As the nouns with a big hierarchy tend to have a larger value for $|cd_{ij}|$, the weighted sum of common concepts is normalized in respect with the dimension of the noun hierarchy. Since the size of a hierarchy grows exponentially with its depth, we used the logarithm of the total number of descendants in the hierarchy (i.e. $log(descendents_j)$).

3. We also took into consideration and have experimented with a few other metrics, but after running the program on several examples, the formula from Algorithm 2 provided the best results.

5.2.2 An Example

As an example, let us consider the verb-noun collocation *revise law*. The verb *revise* has two possible senses in WordNet 1.6 and noun *law* has seven senses.

First, we applied Algorithm 1 and searched the Internet using Alta Vista, for all possible pairs V-N that may be created using *revise* and the words from the similarity lists of *law*. The following ranking of senses was obtained: *law#2*(2829), *law#3*(648), *law#4*(640), *law#6*(397), *law#1*(224), *law#5*(37), *law#7*(0), where the number in the parentheses indicates the number of hits. By setting the threshold $t = 2$, we keep only sense #2 and #3. The notation $\#i/n$ means sense i out of n possible senses given by WordNet.

Next, Algorithm 2 is applied to rank the four possible combinations (two for the verb times two for the noun). The results are summarized in Table 2: (1) $|cd_{ij}|$ - the number of common concepts between the verb and noun hierarchies; (2) $desc_j$ the total number of nouns within the hierarchy of each sense n_j; and (3) the conceptual density C_{ij} for each pair $n_i - v_j$ derived using the formula presented above.

	$\|cd_{ij}\|$		$desc_j$		C_{ij}	
	n_2	n_3	n_2	n_3	n_2	n_3
v_1	5	4	975	1265	0.30	0.28
v_2	0	0	975	1265	0	0

Table 2: Values used in computing the conceptual density and the conceptual density C_{ij}

The largest conceptual density $C_{12} = 0.30$ corresponds to $v_1 - n_2$: $revise\#1/2 - law\#2/5$. This combination of verb-noun senses also appears in SemCor, file br-a01.

5.2.3 Evaluation and comparison with other methods

The overall results using Algorithm 1 followed by Algorithm 2 on the 384 pairs of words are shown in Table 3. By comparing Table 3 with Table 1 one can see the contribution of Algorithm 2 beyond Algorithm 1.

	top 1	top 2	top 3	top 4
noun	86.5%	96%	97%	98%
verb	67%	79%	86%	87%
adjective	79.8%	93%		
adverb	87%	97%		

Table 3: Final results obtained for 384 word pairs using both algorithms.

To our knowledge, there is only one other method, recently reported, that disambiguates unrestricted nouns, verbs, adverbs and adjectives in texts [Stetina et al. 1998]. The method uses WordNet and attempts to exploit sentential and discourse contexts and is based on the idea of semantic distance between words, and lexical relations. A comparison between the results reported in that paper and our results is shown in Table 4. Both methods are compared against the baseline, i.e. the occurrences of the first senses from WordNet.

	Baseline	Stetina	Yarowsky	Our method
noun	80.3%	85.7%	93.9%	86.5%
verb	62.5%	63.9%		67%
adjective	81.8%	83.6%		79.8%
adverb	84.3%	86.5%		87%
AVERAGE	77%	80%		80.1%

Table 4: A comparison with other WSD methods.

There are several very accurate statistical methods such as the one presented in [Yarowsky 1995], but their disadvantage is that they disambiguate only one part of speech (nouns in this case), and focus only on a few words due to the lack of training corpora.

For applications such as query expansion in information retrieval the method presented here has the additional advantage that it may consider the first two senses for each word, in which case the average accuracy is 91% (from Table 3).

6 Query expansion

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 for each word of 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 words that belong to the same synonym set.

Consider, for example the word *activity*. There are 7 senses in WordNet for this word. The synset for the first sense includes two other synonyms *action* and *activeness*. The similarity list that we can now create for this sense of the word is:
$W = \{ \textit{action, activity, activeness} \}$.

[Voorhees 1998] investigated the efficacy of expanding a query for search in large text collections. She uses WordNet and experiments with 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. Her results have shown that there are no significant differences between the precision obtained while using the four expanding strategies.

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 AND W_2 AND ... AND W_n
(2) W_1 NEAR W_2 NEAR ... NEAR W_n
In most 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 engine failed to rank them in the top of the list. Thus, we sought to find a new operator that filtered out many irrelevant texts.

7 Post-processing with a new operator

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 a more restrictive operator. For this second phase, we propose the following additional operator:

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 consecutive paragraphs, where n is a positive integer. The parameter n selects the number of paragraphs, thus controling the size of the text retrieved from a document considered relevant. 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].

In order to apply this new operator, the documents gathered from the Internet have to be segmented into sentences and paragraphs. Separating a text into sentences proves to be an easy task, one could just make use of the punctuation to solve this problem. However, the paragraph segmentation is much more difficult, and this is due 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 and (3) paragraph indentations.

8 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 does an average salary worker pay in the United States?''`

The linguistic processing module identified the following keywords:
x_1 =(`tax`), pos = noun, sense #1/1
x_2 =(`average`), pos = adjective, sense #4/5
x_3 =(`salary`), pos = noun, sense #1/1
x_4 =(`the United States`), pos = noun, sense #1/2
x_5 =(`worker`), pos = noun, sense #1/4
x_6 =(`pays`), pos = verb, 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.

These keywords are the input for the next step of our system. Using the similarity relation encoded in the WordNet synsets, it yields the following six similarity lists:
W_1 = {`tax, taxation, revenue enhancement`}
W_2 = {`average, intermediate, medium, middle`}
W_3 = {`salary, wage, pay, earnings, remuneration`}
W_4 = {`United States, United States of America, America, US, U.S., USA, U.S.A.`}
W_5 = {`worker`}
W_6 = {`pay`}

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 5 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 AND W_2 AND W_3 AND W_4 AND W_5 AND W_6	49,182
2	W_1 AND (W_2 NEAR W_3) AND W_4 AND W_5 AND W_6	9,766
3	W_1 NEAR (W_2 NEAR W_3) AND W_4 AND W_5 AND W_6	976
4	W_1 NEAR W_2 NEAR W_3 NEAR W_4 NEAR W_5 NEAR W_6	1(no)
5	W_1 AND {`average` W_3} AND W_4 AND W_5 AND W_6	9,045
6	W_1 NEAR {`average` W_3} NEAR W_4 NEAR W_5 NEAR W_6	0

Table 5: Queries with various combinations of operators

The ranking provided by the AltaVista is of no use for us here. None of the ten leading documents in any category provides the desired information. The only document fetched by Query 4 is equally irrelevant:

....The proposed tax cut, and the bigger one promised for next year, if enacted, will be paid for by the Social Security wage taxes of middle and low-income workers of America. Employees have been

> willing to pay these taxes because of the promise of guaranteed Social Security retirement benefits. This Republican tax bill is a betrayal of the low and middle-income workers. The unfairness of these proposals is breath taking.

An analysis of the results in the table above indicates that there is a gap in the volume of documents retrieved with the AltaVista operators. For instance using only the AND operator (Query 1) 49,182 documents were obtained, but the NEAR operator (Queries 4 and 6) produced only one output, an irrelevant output, respectively 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.

9 Results

[Leong 1997] classifies the queries as *concrete* and *abstract.* In this classification, the concrete queries are defined as queries based on more specialized knowledge of a domain, while the abstract queries are those based on descriptions.

For the purpose of testing our system, we considered 50 questions derived from the descriptive section of each of the 50 topics in the TREC-6 set and 50 real questions used by users to search the Internet. Let us denote these sets as the *TREC* set, respectively the *REAL* set. In our experiment, the *REAL* queries posed by users are usually concrete queries, while the TREC topics lead to abstract queries. As will be seen, there is a large difference in the system performance for these two types of questions.

Table 6 presents five randomly selected questions from the *TREC* set and five questions from the *REAL* set, together with the results obtained.

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

A summary of the results for the 100 questions used to test our system is presented in Table 7. First, it is shown the *number of documents retrieved* for an average *TREC* and *REAL* question. Naturally, the query extension determined an increase in the number of documents by a factor varying from 1 (meaning equal number of documents retrieved for

Question	AND x_i	NEAR x_i	AND w_i	NEAR w_i	Paragraph w_i
TREC questions					
Which are some of the organizations participating in international criminal activity?	27,716 0	3 1	48,133 0	5 1	6 1
Is the disease of Poliomyelitis (polio) under control in the world?	9,432 1	13 3	10,271 2	15 3	40 11
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
Which are some of the endangered mammals?	32,133 0	6,214 1	32,133 0	6,214 1	150 80
Which are the most crashworthy, and least crashworthy, passenger vehicles?	246 0	5 1	260 1	5 1	15 6
REAL questions					
Where can I find cheap airline fares?	1,360 2	3 3	2608 2	35 5	61 34
Find out about Fifths disease.	2 0	0 0	30 1	0 0	10 1
What is the price of ICI?	4503 0	202 0	10221 0	575 1	117 10
Where can I shop online for Canada??	36049 0	858 1	36049 0	858 1	15 8
What are the average wages for event planners?	6 1	0 0	70 0	0 0	6 6

Table 6: A sample of the results obtained for randomly selected questions from the *TREC* and the *REAL* sets.

both the unextended and extended queries) to 32. The number of paragraphs returned by the new operator is only 26 respectively 48. Moreover, instead of providing full documents, our system identifies the portion of the document where the answer is, which is another reduction factor not captured in the table above.

Next, the *precision*, or the ratio between the number of relevant documents retrieved over the total number of documents retrieved is shown. Since it is impractical to search for the relevant documents among all the documents retrieved by AltaVista for a query, we have considered only the relevant documents in the first ten ranked documents. But, in the case of PARAGRAPH, since the number of paragraphs retrieved is small, the precision was considered over the entire set. With the PARAGRAPH operator, the actual precision reaches 43% for the *TREC* questions and 27.7% for the *REAL* questions. The difference can be explained by the fact that users tend to use short questions, which leads to a very large number of documents found and makes much harder the task of retrieving relevant information.

The biggest gain, however, is in the system *productivity*, the percentage of the questions answered correctly; 90% of the questions from the *TREC* set and 66% from the *REAL* set were answered. This is a significant improvement over the current technology.

In general, it is difficult to compare the performance of Question Answering systems since the range of the questions is so broad. Other systems implemented so far for the *REAL* type of questions attempt to retrieve answers for narrow questions or operate in narrow domains. For example [FindLaw] is designed to find legal resources on Internet. The system described in [Burke, Hammond et al. 1995] uses the files of "Frequently Asked Questions"

	AND	NEAR	AND	NEAR	PARAGRAPH
Question	x_i	x_i	w_i	w_i	w_i
Number of documents retrieved					
Average question from the *TREC* set	7,746	258	25,803	332	26.04
Average question from the *REAL* set	13,510	1,843	28,715	3,003	48.95
Precision					
Average question from the *TREC* set	1.6%	4.8%	4.4%	8.8%	43%
Average question from the *REAL* set	6.3%	12.43%	6.09%	13.65%	27.7%
Productivity					
Average question from the *TREC* set	36%	44%	20%	36%	90%
Average question from the *REAL* set	30%	42%	28%	48%	66%

Table 7: Summary of results for 50 questions from the TREC collection and 50 questions from the frequently asked queries on the Internet.

(FAQs) associated with many Usenetgroups.

For the results obtained during the TREC tests, a comparison can be made with the work described in [Voorhees 1994], even this works is related with the task of retrieving information on very large collections of texts, rather than on the Internet. 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.

10 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].

Limitations.
Even the method proposed in this paper is able to improve the precision in finding correct answers to *abstract* or *concrete* queries, still there are particular questions for which no relevant answers could be found. Very short but broad questions usually lead to a very large number of documents in which it is hard to find relevant information. An example of such a question is: *"What is the land like in Costa Rica?"* The queries formed with boolean operators extracted 26,304 documents when the AND operator was used, respectively 1956 documents when the NEAR operator was used, with no relevant information in top 10 ranked documents. The query expansion phase brought no modifications in the structure of the query, as both *"land"* and *"Costa Rica"* have no synonyms. Using the PARAGRAPH operator, 713 paragraphs were retrieved, too many to be useful for finding a relevant answer.

At the other extreme, questions with very specialized terms also lead to no results. For the question *"Where can I find a cartoon depicting the Sugar Act of 1764?"*, we obtained zero documents using an AND-query, and zero documents with a NEAR-query. The query expansion phase increased the former number to 15, but the PARAGRAPH operator could not find any relevant information.

Extensions.
An easy extension of this approach is to restrict the output to several sentences instead of paragraphs. The SENTENCE n (... similarity lists ...) 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 a single, sometimes complex sentence. This operator works well for queries whose answers are single datum or a list of data found verbatim in text. The parameter n indicates the number of sentences allowed; by default it is set to one.

Also, a more flexible NEAR search could be implemented with a new operator SEQUENCE ($W_1 d W_2 d .... W_n$), 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 requires that the sequence of the words in the similarity list be maintained as specified.

There are several other posible ways of improving the web search not discussed in this paper. One such a possibility is to index words by their WordNet senses, so called *semantic* or *conceptual indexing*. This of course implies some on-line parsing and 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 allowing 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 collar worker*, *stock market*, *mortgage interest rate* 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 Artifical 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

[Buckley et al. 1994] Buckley, C.; Salton, G.; Allan, J. and Singhal, A. Automatic Query Expansion Using SMART: TREC 3. The Third Text REtrieval Conference (TREC-3), NIST Special Publications, Edited by D.Harman, 69-81.

[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 Artifical Intelligence Conference, Fall Symmposium, "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.

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

[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 Artifical Intelligence Conference, Fall Symposium "AI Applications in Knowledge Navigation & Retrieval", 65-71, Cambridge, MA, 1995

[Ishikawa, Satoh and Okumura 1997] Ishikawa, K.; Satoh, K. and Okumura, A. Query Term Expansion based on Paragraphs of the Relevant Documents. The Sixth Text REtrieval Conference (TREC-6), NIST Special Publications, Edited by E.M.Voorhees and D.Harman, 577-585.

[Katz 1997] Katz, B. From Sentence Processing to Information Access on the World Wide Web, Proceedings of the American Association for Artifical Intelligence Conference, Spring Symposium, "NLP for WWW", 77-86, Stanford University, CA, 1997

[Leong 1997] Leong, M.K. Concrete Queries in Specialized Domains: Known Item as Feedback for Query Formulation. The Sixth Text REtrieval Conference (TREC-6), NIST Special Publications, Edited by E.M.Voorhees and D.Harman, 541-550.

[Lu and Keeffer 1994] Lu, X.A. and Keefer, R.B. Query Expansion/Reduction and its Impact on Retrieval Effectiveness The Third Text REtrieval Conference (TREC-3), NIST Special Publications, Edited by D.Harman, 231-240.

[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 Artifical 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, Ing. 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.

[Stetina et al. 1998] Stetina, J., Kurohashi, S., and Nagao, M. General word sense disambiguation method based on a full sentential context. *Usage of WordNet in Natural Language Processing, Proceedings of COLING-ACL Workshop*, Montreal, Canada, July 1998.

[TREC 1997] Text REtrieval Conference
http://trec.nist.gov 1997

[Voorhees 1994] Voorhees, E.M. Query Expansion using Lexcial-Semantic Relations Proceedings of the 17th Annual International ACM SIGIR, Conference on Research and Development in Information Retrieval, 61-69, Dublin, Ireland, 1994.

[Voorhees 1998] Voorhees, E.M. Using WordNet for Text Retrieval WordNet - An Electronic and Lexical Database, Christiane Fellbaum editor, MIT Press, 1998, pp 285-303.

[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

[Yarowsky 1995] Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association of Computational Linguistics*