Automatic Word Similarity Detection for TREC 4 Query Expansion

Susan Gauch and Meng Kam Chong
sgauch@eecs.ukans.edu
Electrical Engineering and Computer Science
University of Kansas

ABSTRACT

Accessing online information remains an inexact science. While valuable information can be found, typically many irrelevant documents are also retrieved and many relevant ones are missed. Terminology mismatches between the user's query and document contents is a main cause of retrieval failures. Expanding a user's query with related words can improve search performance, but the problem of identifying related words remains.

This research uses corpus linguistics techniques to automatically discover word similarities directly from the contents of the untagged TREC database and to incorporates that information in the PRISE information retrieval system. The similarities are calculated based on the contexts in which a set of target words appear. Using these similarities, user queries are automatically expanded, resulting in conceptual retrieval rather than requiring exact word matches between queries and documents.

1. INTRODUCTION

Expanding a user's query with related terms can improve search performance. Relevance feedback systems, where related terms come from the contents of user-identified relevant documents, have been shown to be quite effective (Harman 1992). Our earlier work showed that an expert system which automatically reformulated Boolean queries by including terms from an online thesaurus was able to improve search results (Gauch and Smith 1991; Gauch and Smith 1993) without requiring relevance judgments from the user. Some systems (Anick, Brennan et al. 1990) present related terms to the user and allow them to selectively augment the query. However, the latter two approaches require the presence of an online thesaurus whose words closely match the contents of the database.

Where can such a thesaurus come from? In some cases, it is hand-built (Gauch and Smith 1991), a time-consuming and ad hoc process. In other cases, the thesaurus is an online version of a published thesaurus or semantically coded dictionary (Liddy and Myaeng 1993). However, an online published thesaurus or dictionary will have serious coverage gaps if used for technical domains which have their own distinct sublanguages. Because of ambiguity, this type of thesaurus may also be difficult to use with a database of general English documents because they show all possible classifications for a word when only one or a few senses may be actually present in the database.

Our system to automatically discovers related words directly from the contents of a textual database and incorporates that information in a traditional information retrieval system. We modified and applied one particular techniques from the field of corpus linguistics which seemed particularly well-suited for this task. HNC's MatchPlus system (Gallant, Hecht-Nielsen et al. 1993) has a similar approach, however, they use neural networks to identify features which are used to index documents rather than using the words themselves. In contrast, we index documents by their words and identify related words which can be used for query expansion. With our approach, it is possible to provide query expansion on top of pre-indexed collections.

2. SYSTEM ARCHITECTURE

Our goal is to incorporate the results of the corpus analysis into an existing retrieval engine. For this purpose, we evaluated three freely available text retrieval engines: freeWAIS, SMART and PRISE. These three retrieval engines are all vector space model which use inverted file database structures. Each retrieval engine was evaluated on three document collections, a database of biology paper abstracts, the standard CACM rest collection and the TREC3 database. Based on the results, the PRISE system was selected for our TREC4 entry. It was modified to allow it to expand queries based on the similarity matrices, search the database with the expanded queries, and return the top 1000 documents for each query.

We participated in category B, which is evaluated based on two collections: the Wall Street Journal (WSJ) and the San Jose Mercury (SJM). Combined, the databases are 0.5 GB in size and contain 164777 documents. Indexing the database took approximately 11 hours on a shared Sun SPARC 10. Both databases are in SGML format, which is the input format for PRISE. The stemming function of the PRISE system is turned off, since the automatic query expansion phase will introduce words which share a stem, if their usage in the database is similar enough.

3. CORPUS LINGUISTICS TECHNIQUE

Methods that work with entirely untagged corpora have recently been developed which show great promise (Brill and Marcus 1992; Finch and Chater 1992; Hearst, 1992, Myaeng and Li 1992; Schütze 1992). Using a much more fine-grained approach than traditional automatic thesaurus construction techniques, word-word similarities are automatically calculated based on the premise that words which occur in similar contexts are similar. These techniques are particularly useful for specialized text with specialized vocabularies and word-use, for which there are no adequate online dictionaries. They are also appropriate for general English corpora since a general online dictionary may show many senses for a common word where only one or a few actually are used in a given corpus.

We have modified a corpus linguistics approach (Finch and Chater 1992) that takes into account both the relative positions of the nearby context words as well as the mutual information (Church and Hanks 1990) associated with the occurrence of a particular context word. We have applied this to a sample of the TREC4 database to calculate *a priori* the similarities of a subset of the words in the database, called the target words.

3.1 Similarity Calculation

Similar to (Finch and Chater 1992), the context vector for a word (the *target word*) is a concatenation of the vectors describing the words observed in the preceding two positions and the following two positions (the *context positions*). Each position is represented by a vector corresponding to the occurrence of the 150 highest frequency words in the corpus (the *context words*), giving a 600-dimensional vector describing the context. Initially, the counts from all instances of a word form w_i are summed so that the entry in the corresponding context word position in the vector is the sum of the occurrences of that context word in that position for the corresponding target word form; it is the joint frequency of the context word.

Consider an example in which there are only five context words, {"a", "black", "dog", "the, "very"} and two sentences containing the target word "dog:

(1) The black dog barked very loudly.

(2) A brown dog barked very loudly.

Sentence	Context Position	Observed Word	Context Vector
1	-2	"The"	(0, 0, 0, 1, 0) 4th context word
	-1	"black"	(0, 1, 0, 0, 0) 2nd context word
	+1	"barked"	(0, 0, 0, 0, 0) not a context word
	+2	"very"	(0, 0, 0, 0, 1) 5th context word

Table 1. The context vectors for each of the 4 context positions around the occurrence of the target word "dog" in sentence 1.

The context vector for "dog" in sentence 1 is formed by concatenating the context vectors for each of the 4 context positions:

(0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)

Similarly, the context vector for "dog" in sentence 2 would be:

(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)

and the combined vector for the word "dog" would be:

(1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2)

Using 150 context words, 600-dimensional context vectors are created. Subsequently, 600-dimensional vectors of mutual information values, *MI*, are computed from the frequencies as follows,

$$MI(cw) = \log_2\left(\frac{Nf_{cw}}{f_c f_w} + 1\right)$$

This expresses the mutual information value for the context word *c* appearing with the target word *w*. The mutual information is large whenever a context word appears at a much higher frequency, f_{CW}, in the neighborhood of a target word than would be predicted from the overall frequencies in the corpus, f_C and f_W. The formula adds 1 to the frequency ratio, so that a 0 (zero) occurrence corresponds to 0 mutual information. When the mutual information vectors are computed for a number of words, they can be compared to see which words have similar contexts. The comparison we chose is the inner product, or cosine measure, which can vary between -1.0 and +1.0 (Myaeng and Li 1992).

Finally, to make the identification of the most highly similar terms to a given term more efficient, an auxiliary file is produced *a priori* from the similarity matrix. It stores, for each target word, the words and similarity values for all words with similarity above a given threshold. This is called the similarity lists.

3.2 Preprocessing the Database

For use by the corpus analysis program, the TREC4 database is sampled randomly and uniformly to get a representative sample (roughly 15%) of the whole database. The resulting WSJ sample was 29 MB. Words in the sample range in frequency from 1 to 280,004. There are 102,912 distinct words and 79619 words have frequency less than 6. The corresponding SJM sample is 33 MB and contains words with frequency from 1 to 334,507. There are 113,240 distinct words and 75,602 words have frequency less than 6. During preprocessing, the required fields of a document are extracted and all capitalized tokens are put in lower case. Then, a sentence tagging algorithm has been implemented to identify sentence breaks. Furthermore, the sentences are tokenized by separating the comma, full stop, semicolons etc. from the words.

We then create a file containing the list of words in each sample, sorted by frequency. From this list, the corpus program automatically selects a set of target words and a set of context words. These are used as input to the similarity calculation phase (details in section 3.2). The target words are those which will have their similarities calculated. For efficiency reasons we want to limit the number of words studied to around 10,000. It takes about 10 hours per sample to calculate the 10,000 x 10,000 similarity matrix. However, only words in the target words will be able to be expanded should they appear in a query. Therefore, it is important to select a set of target words that will best match the content bearing words which appear in queries. To help us select our target words well, we analyzed the distribution of the words that used in the 150 TREC3 queries.

Based on our analysis, we found that 50% of the query words are the words which have frequency count greater than 2000 in both databases. These words are like “the”, “an”, “often” and etc. Those are the words that do not contain a lot of information, but do give a lot of context information. So, these words have been selected as the context words. Those words which fall between frequency count 100 and 2000 are the words that appear to be information-bearing yet are frequent enough to be studied statistically. Therefore, these are the words that have been selected as the target words. Thus, we select the target words as those whose frequencies are between 0.03% and 0.8% of the most frequent word in the sample. Context words are the words which have frequency count greater than 0.8% most frequent word.

3.3 Corpus Analysis Modification for Large Databases

In the original program, all information was stored in memory. We modified the program so that information is stored into randomly accessible binary files. This modification does improve the usage of memory and the space complexity becomes n versus n^2 in the original version. However, some of the speed of the program has been sacrificed. Another modification was to create an auxiliary file which stores, for each target word, a sorted list of all words whose similarity is greater than a given threshold. These *similarity lists* are also written in binary format, which allows reduces the memory usage of the modified retrieval engine.

4. TREC4 EXPERIMENT

4.1 Matrix Selection Algorithm

Since there are two databases (WSJ and SJM) in this project, two similarity matrices are generated, one for each database. We chose to do this because each of the databases has a different domain of interests, resulting in different word usage. However, this means that it is very important to select the correct matrix to expand a given query. Thus, for each query, we first calculate which database it best matches using a simple calculation based on the sum of the frequencies of the query words in each database. Whichever database maximizes the sum has its corresponding similarity matrix selected to expand the query.

This technique was tested with queries 101-200 from TREC3. The results are promising. Queries 101-150 are the queries that used to query both databases. The simple calculation assigned 28 of these 50 queries are to the SJM database. In contrast, queries 151-200 are the queries intended for the WSJ database. By applying the same technique to these 50 queries, 41 queries out of the 50 queries are identified as be related to database WSJ.

4.2 PRISE Retrieval Engine Modification

PRISE has been modified at the point just before a query is passed to the search engine. At that point, the query is expanded with the appropriate similarity matrix. The modified PRISE uses five files besides in addition to the regular inverted files. These are: the two similarity lists, frequency counts for the two databases, and a “threshold” file. The threshold file specifies the minimum similarity score necessary for a word to be included by query expansion. In addition, the display routine has been modified to display the similarity scores of a document - query matches

4.3 Query Expansion and Comparison of Matrices

Once the three systems were evaluated and the PRISE system chosen, we ran a series of experiments with the TREC3 queries to tune the corpus analysis program. There are four main parameters that can be adjusted: the list of target words, the list of context words, and the window size (the window around target words that is used to characterize the contexts in which they appear), and the similarity threshold that is used during query expansion. Due to time limitations, we were unable to run as many experiments as we would have liked, only evaluating the effects of changing the window size and similarity threshold used during query expansion. There were 4817 and 5205 target words for the WSJ and SJM, respectively. One hundred ninety-eight context words were used for the SJM database, and 261 context words for the WSJ database. The following table shows the 11 point average for the experiments:

Window Size	**Threshold**	**11 point average**
5	0.30	0.0648
5	0.45	0.0820
5	0.55	0.1064
5	0.57	0.1059
7	0.30	0.0768
7	0.38	0.0802
7	0.40	0.1002
7	0.43	0.1070
7	0.44	0.1068
7	0.45	0.1056
7	0.47	0.1059
Without Expansion		0.1037

Table 2. The 11 point averages based on different context window sizes and query expansion similarity thresholds.

Based on these results, for TREC 4, similarity matrices were constructed using a window of 7 and 0.43 was used as the threshold during query expansion.

4.4 TREC4 Results

The TREC4 queries were shorter than those in the previous TRECs. Thus, I expect that query expansion will be even more important. Here is a table summarizing our results compared to the entries in the ad hoc category:

Topic	# Rel.	Best	Median	Worst	KU1
202	178	66	62	52	66
203	20	8	5	1	5
204	140	53	35	5	24
205	63	5	4	2	5
206	16	2	2	0	0
207	43	33	30	27	33
208	26	10	9	2	8
209	27	13	7	2	2
210	5	2	2	0	0
211	124	21	19	15	21
212	78	39	36	28	32
213	8	1	1	1	1
214	1	1	0	0	1
215	83	41	34	20	32
216	23	11	9	8	8
217	14	6	3	2	3
218	40	16	11	8	16
219	64	22	18	11	17
220	3	3	2	2	3
221	83	35	34	29	31
222	16	9	8	7	7
223	97	52	50	43	52
224	79	25	19	12	19
225	38	14	13	10	13
226	66	8	2	0	8
227	115	33	27	12	25
228	32	8	7	4	7
229	7	5	3	0	2
230	68	22	14	2	10
231	4	4	4	3	4
232	4	0	0	0	0
233	64	4	3	2	3
234	6	4	3	3	4
235	59	18	12	3	12
236	0	0	0	0	0
237	121	38	33	21	33
238	70	11	5	1	11
239	32	7	5	4	4
240	89	24	6	3	3
241	33	2	2	1	1
242	18	6	3	0	2
243	32	12	10	5	10
244	170	62	49	26	26
245	29	9	6	3	5
246	118	29	25	8	23
247	14	7	7	3	3
248	20	5	5	0	5
249	10	2	0	0	2
250	30	12	11	8	12

4.5 Discussion

The results need to be analyzed to compare the results with expansion to those produced by the unmodified PRISE system. I suspect that the quality of our results at this time are primarily due to the quality of the PRISE system. The main problem is that our target words (those which can be expanded) do not include the important words in the query. For example, on topic 203, we retrieved only 5 relevant documents out of a possible 20, which was the median, whereas the best system retrieved 8. This was a query which was not much expanded by our system:

Original: *What is the economic impact of recycling tires?*

After expansion: *what 1.000000 is 1.000000 the 1.000000 economic 0.316406 political 0.156178 financial 0.154311 nuclear 0.126436 civil 0.124980 foreign 0.121690 impact 1.000000 of 1.000000 recycling 1.000000 tires 1.000000*

The only word expanded is economic, which was expanded with *political, financial, nuclear, civil,* and *foreign* with decreasing similarity weight. While these are reasonably similar words, the key words in the query, *recycling* and *tires*, were not target words and thus were not expanded at all.

Our next tasks are to do a better job of identifying the appropriate set of target words and further tuning the corpus analysis program to improve our results.

BIBLIOGRAPHY

Anick, P.G., Brennan, J.D., Flynn, R.A., Hanssen, D.R., Alvey, B., and Robbins, J.M. (1990). A Direct Manipulation Interface for Boolean Information Retrieval via Natural Language Query. Proc. 13th Ann. International ACM SIGIR Conf., (pp. 135-150). Brussels, Belgium: ACM Press.

Brill, E., & Marcus, M. (1992). Tagging an Unfamiliar Text with Minimal Human Supervision. In AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language (Working Notes), (pp. 10-16). Cambridge, MA.

Church, K. W., & Hanks, P. (1990). Word Association Norms, Mutual Information and Lexicography. Computational Linguistics, 16(1), 22-29.

Finch, S., & Chater, N. (1992). Bootstrapping Syntactic Categories Using Statistical Methods. In W. Daelemans & D. Powers (Ed.), Proc. 1st SHOE Workshop, (pp. 229-235). Tilburg U., The Netherlands.

Gallant, S.I., Hecht-Nielsen, R., Caid, W., Qing, K., Carleton, J., and Sudbeck, C.D. (1993). HNC's MatchPlus System, 1st Text Retrieval Conf. (TREC-1), (pp. 107-111). NIST #500-207.

Gauch, S., & Smith, J.B. (1993). An Expert System for Automatic Query Reformulation. J. of the Amer. Society of Inf. Sci., 44 (3), 124-136.

Gauch, S., & Smith, J.B. (1991). Search Improvement via Automatic Query Reformulation. ACM Trans. on Information Systems, 9 (3), 249-280.

Harman, D. (1992). Relevance Feedback Revisited. Proc. 15th Ann. International ACM SIGIR Conf., (pp. 1-10). Copenhagen, Denmark: ACM Press.

Hearst, M.A. (1992) "Automatic Acquisition of Hyponyms from Large Text Corpora," Proc. 14th Intern'l Conf. on Computational Linguistics, Nantes, France, July.

Liddy, E.D., & Myaeng, S.H. (1993). DR-LINK's Linguistic-Conceptual Approach to Document Detection, 1st Text Retrieval Conf. (TREC-1), (pp. 113-129). NIST #500-207.

Myaeng, S. H., & Li, M. (1992). Building Term Clusters by Acquiring Lexical Semantics from a Corpus. In Y. Yesha (Ed.), CIKM-92, (pp. 130-137). Baltimore, MD: ISMM.

Schütze, H. (1992). Context Space. In AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language (Working Notes), (pp. 113-120). Cambridge, MA: