Query Expansion using Lexical-Semantic Relations

Ellen M. Voorhees

ellen@scr.siemens.com

Siemens Corporate Research, Inc.
755 College Road East
Princeton, NJ 08540

Abstract

Applications such as office automation, news filtering, help facilities in complex systems, and the like require the ability to retrieve documents from full-text databases where vocabulary problems can be particularly severe. Experiments performed on small collections with single-domain thesauri suggest that expanding query vectors with words that are lexically related to the original query words can ameliorate some of the problems of mismatched vocabularies. This paper examines the utility of lexical query expansion in the large, diverse TREC collection. Concepts are represented by WordNet synonym sets and are expanded by following the typed links included in WordNet. Experimental results show this query expansion technique makes little difference in retrieval effectiveness if the original queries are relatively complete descriptions of the information being sought even when the concepts to be expanded are selected by hand. Less well developed queries can be significantly improved by expansion of hand-chosen concepts. However, an automatic procedure that can approximate the set of hand picked synonym sets has yet to be devised, and expanding by the synonym sets that are automatically generated can degrade retrieval performance.

1 Introduction

Users of retrieval systems that use word matching as a basis for retrieval are faced with the challenge of phrasing their queries in the vocabularies of the documents they wish to retrieve. This difficulty is especially severe in large, full-text databases since such databases contain many different expressions of the same concept [1]. Yet the ability to retrieve documents from such databases is crucial in a wide range of applications: retrieving documentation in support of a legal case, facilitating the organization and retrieval of correspondence and forms in an office, filtering news feeds for articles of interest, finding relevant passages within the complete manual set of a complex system for the particular problem at hand, etc. One method of easing the user's burden when selecting query words is for the retrieval system to automatically expand the query by adding terms that are related to the words supplied by the user. The new terms can either be statistically related to the original query words (that is, the terms tend to co-occur with one another in documents) or chosen from lexical aids such as thesauri, controlled vocabulary schedules, and the like.

Using statistical relations to expand query vectors is attractive since the the relations are easily generated from the documents at hand, obviating the need for lexical aids, which are expensive to build and maintain. Unfortunately, such methods have had little success in improving retrieval effectiveness when used apart from relevance data [2, 3]. Indeed, Peat and Willett show there are limitations to the effectiveness one can expect from such systems [4]. (Note, however, that methods that exploit statistical relations but do not expand the query, such as Latent Semantic Indexing [5], have been more successful.)

Using lexical aids as a source of related terms has met with some success in small experiments. Salton and Lesk found that expansion by synonyms improved performance but expansion by broader or narrower terms selected from a hierarchical thesaurus was too inconsistent to be generally useful [6]. Wang, Vendendorpe, and Evens found that a variety of lexical-semantic relations improved retrieval performance [7]. However, each of these conclusions was drawn from experiments on very small collections using single-domain thesauri.

This paper examines the utility of query expansion by lexical-semantic relations in a large collection that spans several domains. Queries are expanded using the relations encoded in WordNet [8], a large, general-purpose lexical system built at Princeton University, and are run against the TREC collection [9]. To eliminate the confounding effects of expanding a poor selection of words, the terms that were expanded were chosen by hand. Thus, the results reported here represent an upper bound on the performance to be expected from a completely automatic procedure that uses this expansion strategy. Even in this best-case scenario, the expansion did not improve the effectiveness of queries that were relatively complete at the

start. Less complete queries — queries consisting of a single sentence describing the topic of interest — were significantly improved by the expansion.

2 The Retrieval Environment

This section provides the background necessary to understand the context in which the experiments were carried out. The following section describes the experiments themselves, and the remaining section summarizes the conclusions the data support.

2.1 WordNet

The expansion procedure used in this work relies heavily on the information recorded in WordNet, a manually-constructed lexical system developed by George Miller and his colleagues at the Cognitive Science Laboratory at Princeton University [8]. WordNet's basic object is a set of strict synonyms, called a *synset*. Synsets are organized by the lexical relations defined on them, which differ depending on part of speech. For nouns (the only part of WordNet used in this study), the lexical relations include antonymy, hypernymy/hyponymy (*is-a* relation) and three different meronym/holonym (*part-of*) relations. The *is-a* relation is the dominant relationship, and organizes the synsets into a set of approximately ten hierarchies[1]. Figure 1 shows a piece of WordNet. The figure contains all the ancestors and descendents as defined by the *is-a* relation for the six senses of the noun *swing*. Also shown is that one of the senses, a child's toy, is *part-of* a playground.

2.2 The TREC Collection

The TREC collection is a test collection being produced as a result of the TREC and Tipster workshops [9]. The part of the collection used in this work consists of the approximately 742,000 documents on TREC disks one and two, queries 101-150, and the set of relevance judgements produced after the TREC-2 and Tipster-3 evaluations.

The TREC documents consist of English prose obtained from a variety of sources including newspapers, abstracts of technical papers, and the *Federal Register*. There are some SGML-like tags in the documents to delineate the bibliographic parts of the document (document number, title/headline, author, etc.). Other tags that mark special punctuation in the body of a document were ignored in this work. The documents were indexed completely automatically using the standard SMART indexing routines [10] (i.e., tokenization, stop word removal, and stemming) to produce an inverted index of document vectors.

The text of a TREC query or, in TREC parlance, *topic statement*, is a complex natural-language statement of need as shown in Figure 2. Each topic statement has a set of fields flagged by special markers (the words enclosed in angle brackets). The Narrative field provides a particularly detailed description of what constitutes a relevant document; the Concepts field usually lists words and phrases that the creator of the statement thinks are related to the topic. A shorter version of each topic statement is also available. This shorter version, the Summary Statement, is usually a single sentence describing the search request. The Summary Statement for the topic shown in Figure 2 is the sentence given in the Description field. (The Summary Statement is frequently, but not always, identical to the the Description field.)

For this work, I added a new field to the topic statements: a list of hand-selected WordNet synsets containing nouns germane to the topic. My goal in selecting synsets for a particular topic was to pick synsets that emphasized important concepts of the topic. One aspect of the problem is sense resolution, i.e., selecting the synset that contains the correct sense of an ambiguous original topic word. I did not restrict myself to adding only synsets that contain some original topic word, however, since one purpose of the experiments is to investigate the efficacy of lexical-semantic relations assuming good starting concepts. Instead, the choice of synsets was governed by my understanding of the full topic statment and the fact that the selected synsets would be used to expand the query. I added an average of 2.7 (minimum 0, maximum 6) synonym sets per topic.

Topic 122 shown in Figure 2 provides an example of how synsets were selected for a topic. The topic asks for information about bringing cancer fighting drugs to market. The text never mentions 'pharmaceutical', but I added the synset {*pharmaceutical*}, a child of the synset {*drug*}, to the text. Early experiments demonstrated that expansion worked poorly when synsets with very many children in the *is-a* hierarchy were used. In addition to *pharmaceutical*, {*drug*} has children for many different types of drugs (stimulants, intoxicants, sedatives, etc.) that are not related to cancer-fighting. I chose the more

[1] The actual structure is not quite a hierarchy since a few synsets have more than one parent.

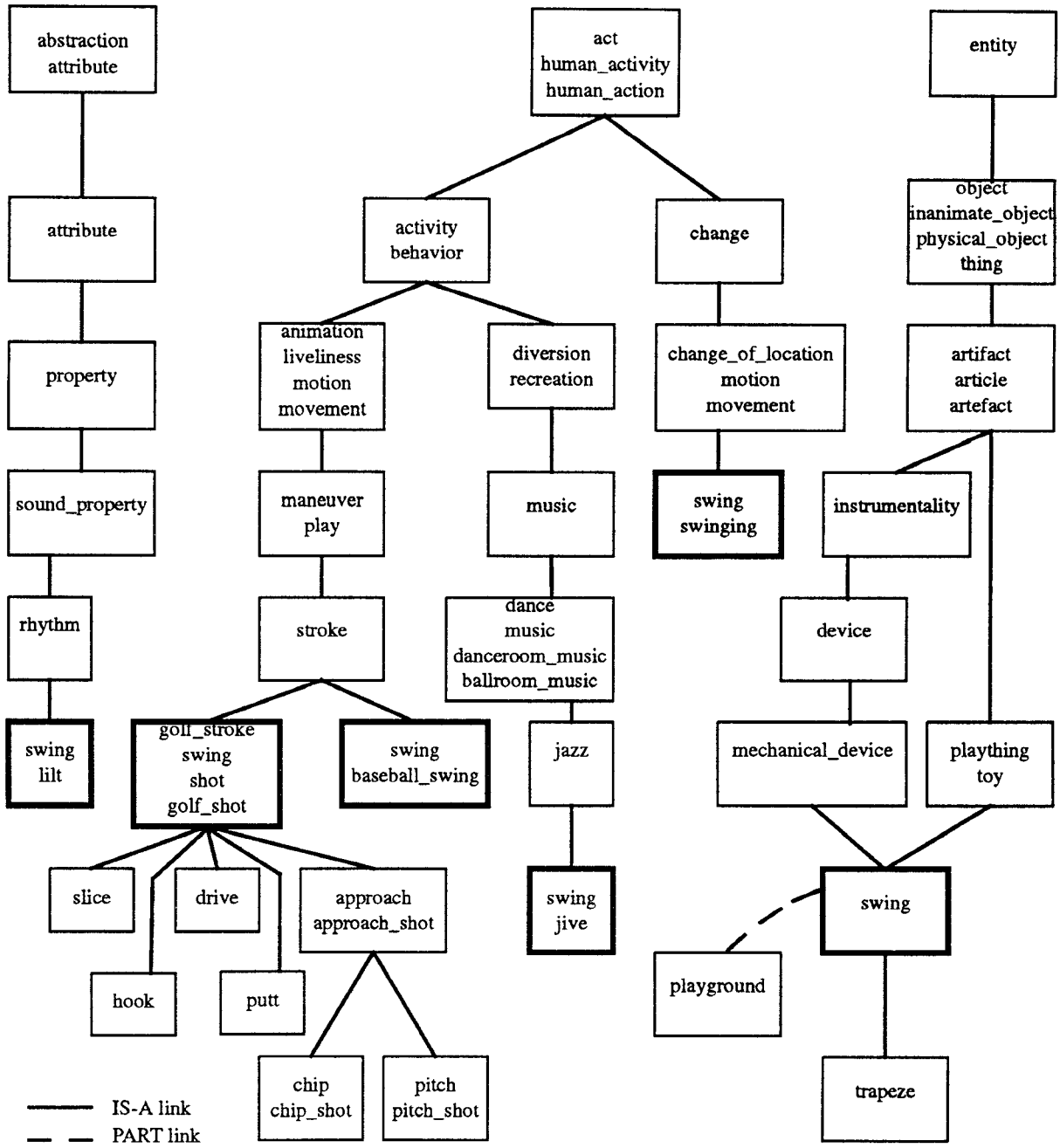

Figure 1. Relations defined for the six senses of the noun *swing* in WordNet.

specific *pharmaceutical* to avoid over-generalizing the expanded query. The complete list of synsets added to topic 122 is {*cancer*}, {*skin_cancer*}, and {*pharmaceutical*}.

Some topics contain important concepts that have no corresponding synset. Occasionally, the missing synset is a gap in WordNet; for example, *toxic waste*, *genetic engineering*, and *sanctions* meaning economic disciplinary measures are not in version 1.3 of WordNet. More often, the important concept was a proper noun or highly technical term that one wouldn't expect to be in WordNet. *SDI* or *Star Wars*, for example, is an important concept for topics 101 and 102 but does not occur in WordNet. Nothing was added to the topic texts for concepts that lacked corresponding synsets in these experiments, although making some provision for them would improve retrieval performance.

2.3 The Expansion Procedure

Once the text of the topics is annotated with synsets, the remainder of the processing is automatic. Selected fields of the original topic statements are indexed using the standard SMART routines. The

```
<dom> Domain: Medical & Biological

<title> Topic: RDT&E of New Cancer Fighting Drugs

<desc> Description:

Document will report on the research, development, testing, and evaluation (RDT&E) of
a new anti-cancer drug developed anywhere in the world.

<narr> Narrative:

A relevant document will report on any phase in the worldwide process of bringing new
cancer fighting drugs to market, from conceptualization to government marketing
approval. The laboratory or company responsible for the drug project, the specific
type of cancer(s) which the drug is designed to counter, and the chemical/medical
properties of the drug must be identified.

<con> Concept(s):

1. cancer, leukemia

2. drug, chemotherapy
```

Figure 2. Topic statement of query 122.

terms derived from these sections are "original query terms". The expansion procedure is invoked when the synonym set section is reached.

Given a synset, there is a wide choice of words to add to a query vector — one can add only the synonyms within the synset, or all descendents in the *is-a* hierarchy, or all words in synsets one link away from the original synset regardless of link type, etc. The expansion procedure is parameterized to facilitate comparing the effectiveness of a variety of these schemes. The parameter set for a given run specifies for each relation type included in WordNet the maximum length of a chain of that type of link that may be followed. A chain begins at each synset listed in the synset section of the topic text and may contain only links of a single type. All synonyms contained within a synset of the chain are added to the query. Collocations such as *change_of_location* in Figure 1 are broken into their component words, stop words such as *of* are removed, and the remaining words are stemmed. The word stems plus a tag indicating the lexical relation through which the stems are related to the original synset are then appended to the original query terms.

As an example of the expansion process, consider the synsets for *swing* shown in Figure 1. If the synset added to the topic is the synset containing *golf_stroke*, and any number of hyponym (child) links may be traversed, then the stems of *golf*, *stroke*, *swing*, *shot*, *slice*, *hook*, *drive*, *putt*, *approach*, *chip*, and *pitch* would be added to the query vector. If hyponym chains are limited to length one, then *chip* and *pitch* would not be added. If the synset added to the topic is the one containing swing meaning plaything and any link type may be followed for one link, then the stems of *swing*, *mechanical*, *device*, *plaything*, *toy*, *playground*, and *trapeze* would be added to the query.

Stems added through different lexical relations are kept separate using the extended vector space model introduced by Fox [11]. Each query vector is comprised of subvectors of different concept types (called *ctypes*) where each ctype corresponds to a different lexical relation. A query vector potentially has eleven ctypes: one for original query terms, one for synonyms, and one each for the other relation types contained within the noun portion of WordNet (each half of a symmetric relation has its own ctype). An original query term that is a member of a synset selected for that query appears in both of the respective ctypes. Similarly, a word that is related to a synset through two different relations appears in both ctypes.

The similarity between a document vector D and an extended query vector Q is computed as the weighted sum of the similarities between D and each of the query's subvectors:

$$sim(D, Q) = \sum_{\text{ctype } i} \alpha_i \times D \cdot Q_i$$

where $\cdot$ denotes the inner product of two vectors, Q_i is the ith subvector of Q, and α_i, a real number, reflects the importance of ctype i relative to the other ctypes. Terms in documents vectors are weighted using the *lnc* weights suggested by Buckley et al. [12]; that is, the weight of a term is set to $1.0 + \ln(tf)$ where tf is the number of times the term occurs in the document and is then normalized by the square root

of the sum of the squares of the weights in the vector (cosine normalization). Query terms are weighted using *ltN*: the log term frequency factor above is multiplied by the term's inverse document frequency, and the weights in the ctype representing original query terms are normalized by the cosine factor. Weights in additional ctypes are normalized using the length computed for the original terms' ctype. This normalization strategy allows the original query term weights to be unaffected by the expansion process and keeps the weights in each ctype comparable with one another.

3 Experiments

3.1 Full Topic Statement

The purpose of this investigation is to determine the efficacy of expanding a query by lexical-semantic relations. Given a set of concepts to be expanded, the effectiveness of an expanded run is dependent on the link types followed during the expansion and the relative weight given to each link type (the α's in the similarity function above), so a variety of different schemes must be tested. Table 1 shows the 11-point average precision value and percent difference over the unexpanded run for different combinations evaluated using the full topic statement (except the "Definitions" field) plus synsets. Four expansion strategies were tried: expansion by synonyms only, expansion by synonyms plus all descendents in the *is-a* hierarchy, expansion by synonyms plus parents and all descendents in the *is-a* hierarchy, and expansion by synonyms plus any synset directly related to the given synset (i.e., a chain of length 1 for all link types). The α for the original terms subvector was usually greater than the α for the other subvectors to reflect the assumption that user-supplied terms are generally superior than automatically added ones. The runs in which the original terms α was less than or equal to another α tested this assumption.

Clearly, the expansion is ineffective: none of the expansion strategies significantly improves the performance of the unexpanded query. Indeed, the difference in performance between an expanded and unexpanded run for individual queries is very small for most expanded runs. Individual query performance differs more for more aggressive expansion strategies (i.e., expanding using longer chains of links and weighting added terms more heavily) but across the set of queries the aggregate performance is worse for aggressively expanded queries.

In an earlier set of experiments, the most effective expanded run was the one that expanded a query synset by any synset directly related to it and had $\alpha = .5$ for all added subvectors [13]. While this combination is not optimal for these queries, it has the advantage of being a straight-forward choice of expansion parameters. Thus, this expansion strategy, which will be called the standard expansion strategy, is used for the experiments described in the next section.

3.2 Less Detailed Topic Statements

Query expansion is a recall-enhancing technique designed to overcome some of the problems caused by differing vocabularies. To test the hypothesis that expansion is unhelpful in the TREC collection due to the very complete problem statement provided by a TREC topic, queries derived from shorter versions of the original topic statements were expanded using the standard expansion strategy. One query set was derived using the Summary Statement plus the Concepts field; another query set was derived using only the Summary Statement. Both new versions used exactly the same set of synsets to expand as did the queries derived from the full topic statement.

Table 2 compares the lengths of the different query vectors. The table contains the mean number of original terms and the mean ratio of additional terms to original terms for each of the different versions of queries: derived from full topic (Full), derived from Summary Statement plus Concepts (SmryCon), and derived from Summary Statement (Summary) only. The mean number of additional terms is the same (17.56) for each version since the same set of synsets is expanded each time.

Figures 3 and 4 contain the retrieval results for the two new versions of the queries. The unexpanded run is the same version of the query as the expanded run with no additional terms added. The base case uses the unexpanded queries derived from the full topic statement. Expansion does not improve the Summary Statement plus Concepts version of the queries, but significantly improves the Summary Statement only version (35% improvement in 11-point average precision). Note, however, that the overall level of effectiveness obtained by the expanded Summary queries is less than the unexpanded full topic queries (39% degradation in the 11-point average precision).

3.3 Automatic Selection of Synsets

Given that short queries have the potential to be significantly improved by expansion, it is necessary to see if the potential can be realized by a completely automatic procedure. While it is possible to present

Unexpanded queries				ave. prec. .3586	% change —
Expansion by synonyms only					
orig terms α	synonyms α				
1	.1			.3614	+0.8
1	.3			.3639	+1.5
1	.5			.3634	+1.3
1	.8			.3629	+1.2
Expansion by synonyms plus all descendents					
orig terms α	synonyms α	descendents α			
1	.1	.1		.3617	+0.9
1	.3	.1		.3639	+1.5
1	.3	.3		.3635	+1.4
1	.5	.1		.3635	+1.4
1	.5	.3		.3637	+1.4
1	.5	.5		.3622	+1.0
1	.8	.1		.3614	+0.8
1	.8	.3		.3612	+0.7
1	.8	.5		.3603	+0.5
Expansion by synonyms plus parents and all descendents					
orig terms α	synonyms α	descendents α	parents α		
1	.1	.1	.1	.3617	+0.9
1	.3	.1	.1	.3640	+1.5
1	.3	.3	.1	.3639	+1.5
1	.3	.3	.3	.3647	+1.7
1	.5	.1	.1	.3639	+1.5
1	.5	.3	.1	.3638	+1.5
1	.5	.3	.3	.3646	+1.7
1	.5	.5	.1	.3624	+1.0
1	.5	.5	.3	.3628	+1.2
1	.5	.5	.5	.3627	+1.1
1	.8	.1	.1	.3622	+1.0
1	.8	.3	.1	.3617	+0.9
1	.8	.3	.3	.3614	+0.8
1	.8	.5	.1	.3605	+0.5
1	.8	.5	.3	.3605	+0.5
1	.8	.5	.5	.3609	+0.6
1	1	1	1	.3511	-2.1
1	2	1	1	.3350	-6.6
Expansion by synonyms plus any directly related synset					
orig terms α	synonyms α	other α			
1	.3	.1		.3629	+1.2
1	.3	.3		.3630	+1.2
1	.5	.1		.3624	+1.0
1	.5	.3		.3620	+0.9
1	.5	.5		.3608	+0.6
1	.3	.5		.3604	+0.5
1	1	1		.3491	-2.7

Table 1. Combinations of expansion strategies and relation weights tested.

users with a list of candidate synsets and have them select the ones to expand, choosing correct synsets is a tedious process, and a poor choice can be worse than not expanding [13].

Figure 5 provides a high-level description of the algorithm developed to select the synsets. The algorithm is based on the observation that the synsets need to represent the correct sense of *important* concepts of the query [14]. Using the same reasoning as is used for inverse document frequency weights [15], importance is approximated by the number of documents in which a query term occurs — a term occurring in more than N documents is not expanded. Sense resolution is approximated by requiring a new term to be related to at least two original query terms before it is included in the expanded query.

A series of retrieval runs using the above procedure on the Summary Statements tested the procedure's effectiveness. The experiments tested different values of N: 70,000, approximately 10% of the collection,

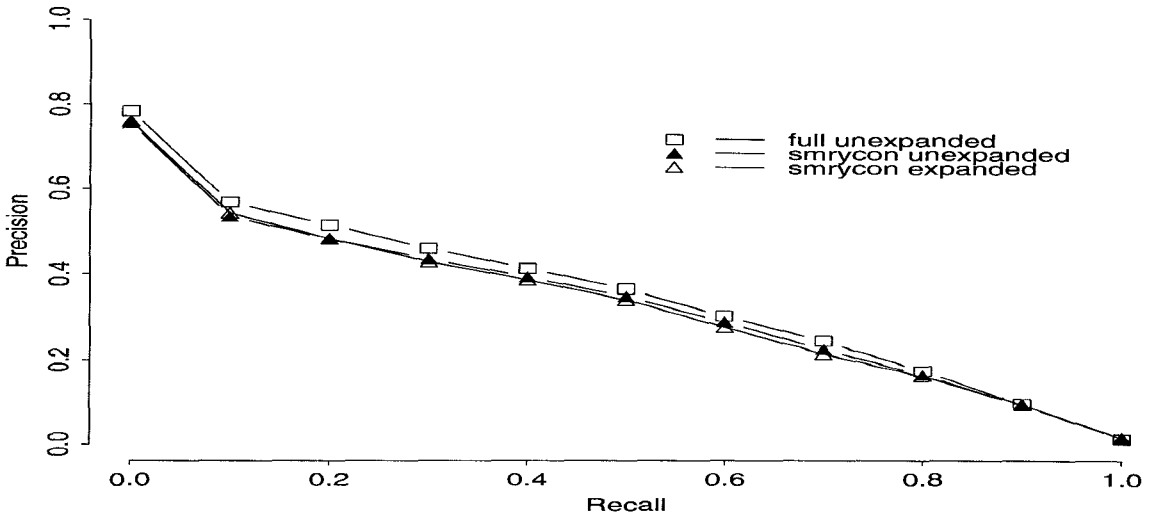

Figure 3. Effectiveness of queries derived from Summary Statement and Concept fields.

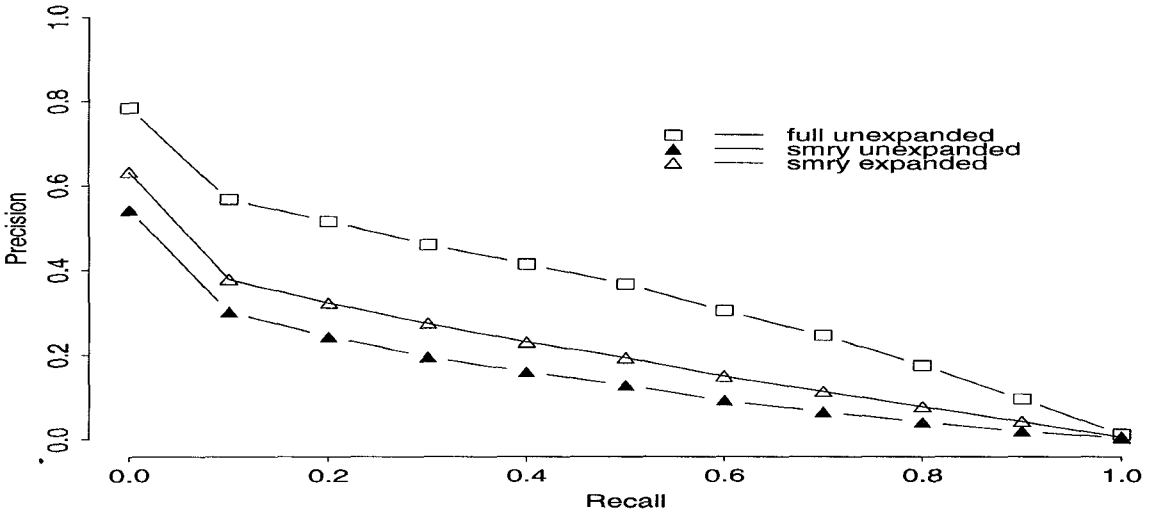

Figure 4. Effectiveness of queries derived from Summary Statement.

and 35,000, approximately 5% of the collection; different limits on the lengths of chains to follow when expanding (all link types were treated identically): 1 and 2; and different α values: .3, .5, and .8. Table 3 shows the 11-point average precision values obtained for these runs. As can be seen, none of the runs materially changes the performance of the unexpanded Summary Statement queries.

Inspection of the queries that resulted from the automatic selection procedure suggests that the requirement that a term appear in two lists is not a good approximation to sense disambiguation. The correct senses of words contained in a short query seldom have common relatives. Instead, the words that

	Full	SmryCon	Summary
Mean number terms	52.54	29.22	11.02
Mean ratio	.36	.77	1.71

Table 2. Length statistics for different versions of queries.

```
for (each query word w) {
        if (w not already expanded and
                        document frequency of w ≤ N ) {
                expand all synsets containing w producing kin list of w
        }
}
for (each relative in the set of kin lists) {
        if (relative occurs in more than 1 list)
                add relative to query vector
}
```

Figure 5. Procedure to automatically select synonym sets to expand.

appear in more than one list are likely to be fairly general terms with more than one sense themselves. For example, since collocations are split into their components during the expansion process, general nouns such as *system* tend to appear in multiple lists.

4 Conclusion

The experiments discussed here demonstrate that expansion by general lexical-semantic relations provides little benefit when a user supplies a detailed query. Since query expansion is a recall-enhancing technique, it is not surprising that longer queries benefit less than shorter queries. However, the longer queries are by no means doing a perfect job of retrieval, and they can be improved by other expansion techniques such as relevance feedback [16]. The success of these other methods suggests that the most useful relations for query expansion are idiosyncratic to the particular query in the context of the particular document collection.

Nonetheless, users frequently do not supply a detailed query. In this case, lexical-semantic relations have the potential to improve an initial query, though this expanded query is unlikely to be as effective as a better formulated user-supplied query. The challenge now lies in finding an automatic procedure that is able to select appropriate concepts to expand.

	ave. prec.	% change
Unexpanded queries	.1634	—
N=70,000; max chain length=1		
$\alpha = .3$	.1627	−0.5
$\alpha = .5$	.1603	−1.9
$\alpha = .8$	.1543	−5.6
N=70,000; max chain length=2		
$\alpha = .3$	.1633	−0.1
$\alpha = .5$	.1557	−4.7
$\alpha = .8$	.1402	−14.2
N=35,000; max chain length=1		
$\alpha = .3$	.1636	+0.1
$\alpha = .5$	.1635	+0.1
$\alpha = .8$	.1639	+0.3
N=35,000; max chain length=2		
$\alpha = .3$	.1645	+0.7
$\alpha = .5$	.1642	+0.5
$\alpha = .8$	.1617	−1.0

Table 3. Effectiveness of expansion strategies on Summary Statement queries when expanding automatically selected synsets.

References

1. David C. Blair and M. E. Maron. Full-text information retrieval: Further analysis and clarification. *Information Processing and Management*, 26(3):437–447, 1990.
2. A. F. Smeaton and C. J. van Rijsbergen. The retrieval effects of query expansion on a feedback document retrieval system. *Computer Journal*, 26:239–246, 1983.
3. C. T. Yu, C. Buckley, and G. Salton. A generalized term dependency model in information retrieval. *Information Technology: Research and Development*, 2:129–154, 1983.
4. Helen J. Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5):378–383, 1991.
5. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
6. G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 143–180. Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1971.
7. Yih-Chen Wang, James Vandendorpe, and Martha Evens. Relational thesauri in information retrieval. *Journal of the American Society for Information Science*, 36(1):15–27, January 1985.
8. George Miller. Special Issue, WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.
9. Donna K. Harman. The first Text REtrieval Conference (TREC-1), Rockville, MD, U.S.A, 4–6 November, 1992. *Information Processing and Management*, 29(4):411–414, 1993.
10. Chris Buckley. Implementation of the SMART information retrieval system. Technical Report 85-686, Computer Science Department, Cornell University, Ithaca, New York, May 1985.
11. Edward A. Fox. *Extending the Boolean and Vector Space Models of Information Retrieval with P-norm Queries and Multiple Concept Types.* PhD thesis, Cornell University, 1983. University Microfilms, Ann Arbor, MI.
12. Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.
13. Ellen M. Voorhees. On expanding query vectors with lexically related words. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, 1993. In press.
14. Ellen M. Voorhees and Yuan-Wang Hou. Vector expansion in a large collection. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 343–351. NIST Special Publication 500-207, March 1993.
15. Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, March 1972.
16. Chris Buckley, James Allan, and Gerard Salton. Automatic routing and ad-hoc retrieval using SMART: TREC 2. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, 1993.