Cooccurrence-Based Vector Expansion

Hinrich Schütze
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
USA
{schuetze}@parc.xerox.com

ISTL Technical Report No. ISTL-QCA-1994-02-01

Abstract

This paper presents a new method for constructing a thesaurus from a text corpus. Each word is represented as a vector in a multi-dimensional space. Entries in the vector are formed from neighbors of the word in the corpus. It is shown that using the thesaurus for vector expansion of queries and documents applied to the TIPSTER reference collection improves performance significantly as compared to a tf.idf baseline.

1 Introduction

Information retrieval systems that define similarity between queries and documents in terms of a weighted sum of matching words (e.g. the vector model as in Salton and McGill 1983) can be defeated by synonymy, i.e. words with the same meaning but dissimilar spelling to words in the query. This problem can be addressed by using a thesaurus which captures synonymy between words. One way to exploit the thesaurus is the method of vector expansion. The standard representation in the vector similarity model is enhanced by adding for each word in the document or query one or more synonyms and other terms that are closely related to it in the thesaurus (see (Sparck Jones 1971) for an early experiment on vector expansion in information retrieval).

This paper presents a new method for constructing a thesaurus for vector expansion. The method is based on lexical cooccurrence. Each word is represented as a vector in a multi-dimensional space. Entries in the vector are formed by counting neighbors of the word in the corpus. For compactness and improved generalization, a singular value decomposition is used to reduce the dimensionality of the thesaurus space (cf. Deerwester et al. 1990).

The following sections discuss previous approaches to thesaurus construction and vector expansion, describe the cooccurrence-based thesaurus, and show that a vector expansion method applied to the TIPSTER collection (Harman 1993) improves performance as compared to tf.idf.

2 Related Work

In order to expand terms with closely related words, one needs a data structure that defines semantic relatedness. In most approaches, this data structure is a thesaurus. Even if a thesaurus is not explicitly computed, one can always conceptualize vector expansion as being based on a thesaurus. Therefore, we will first discuss previous approaches to thesaurus construction and then comment on vector expansion work proper.

Manual thesaurus construction is labor-intensive and therefore inappropriate except for specialized domains[1]. Various automatic methods of thesaurus construction have been proposed in the literature, three of which will be discussed: a so called "first-order" thesaurus based on a hierarchical ordering of noun phrases (Evans et al. 1991), thesauri based on lexicographic databases and thesauri based on occurrence-in-document counts derived from text collections.

Evans et al. (1991) construct a thesaurus from a lattice of complex noun phrases in which subsumption roughly corresponds to the subset relation defined on terms ("intelligence" subsumes "artificial intelligence"). While this method is superior to approaches that treat terms as unanalyzed atoms, there is no notion of semantic similarity of basic terms. For example, it is not clear how the semantic similarity of "astronaut" and "cosmonaut" could be discovered using the subsumption hierarchy. This paper focuses on the semantic similarity of otherwise unrelated basic terms.

If lexicographic databases such as WordNet or Longman's subject codes are used (Voorhees and Hou 1992, Liddy and Paik 1992), then the lexical representations in the thesaurus are often

[1] See (Chute et al.) for an example of the use of a hand-constructed thesaurus in information retrieval

not specific enough for the text collection at hand. For example, "acts" may be expanded with "acts of the apostles" in a legal text collection (cf. Voorhees and Hou 1992). In addition, no information about proper nouns is recorded in dictionaries. Yet proper nouns are often excellent retrieval cues. Therefore, the approach taken here is to derive the thesaurus entirely from the text collection for which the retrieval system is intended.

Crouch 1990 presents such an approach, based on the term discrimination model. Her approach requires both a single-link and complete-link clustering of the document collection, which would be too expensive for a large collection like TIPSTER. Crouch uses the occurrence of terms in documents as the crucial input. This seems problematic for collections with long documents if the goal is to get at properties of individual words. In comparison to document occurrence, lexical cooccurrence is more informative, both qualitatively and quantitatively (cf. Schütze 1992). The qualitative difference is that the fact that two words often occur close to each other is more likely to be significant than the fact that they occur in the same documents. Quantitatively, there are on an order of magnitude more cooccurrence events than occurrence-in-document events in a given document collection. For a word occurring n times in the document collection and for a definition of cooccurrence as occurring in a window of k words, there are nk cooccurrence events, but only n occurrence-in-document events. These considerations suggest that with equal computational effort lexical cooccurrence is the preferred basis of thesaurus construction.

Another important feature distinguishes the thesaurus constructed here from the one presented in (Crouch 1990). While the latter consists of a collection of thesaurus classes, we will construct a continuous space in which each word has its own individual representation as a vector in the space. Any class-oriented or clustering method introduces artificial boundaries that must cut off some words from part of their topical neighborhood. Members from different topics are forced together in large classes. Hence, any class size is problematic, since words are either separated from close neighbors or lumped together with distant terms. In contrast, a continuous space does not force a classification choice, and hence avoids these problems.

Peat and Willett (1991) argue against the utility of cooccurrence-based expansion in principle. They observe that synonyms often do not occur together so that a cooccurrence-based approach may have difficulty to identify synonymy relations. However, although synonyms frequently don't cooccur, they tend to share neighbors that occur with both. For example, "shore" and "coast" share neighbors such as "water", "ship", and "tide". Our thesaurus will make use of common neighbors to define semantic closeness and can thus find synonyms even if they don't cooccur.

A second problem pointed out by Peat and Willett (1991) is that many researchers use measures for defining closeness that will group words according to frequency: it is impossible for a frequent word to have an infrequent neighbor according to these measures. We avoid this difficulty by reducing the dimensionality of the thesaurus space using a singular value decomposition. The reason for the closeness of terms with equal frequency is, roughly, that they have about the same number of zero entries in their term vectors. For a given term, SVD assigns values to all dimensions of the space, so that frequent and infrequent terms can be close in the reduced space if they occur with similar terms.

Ruge (1992) and Grefenstette (1992) use head-modifier relationships to determine semantic closeness. This solution is costly since parsing technology is required to determine head-modifier relations in sentences. It is also unclear to what extent words with similar heads or modifiers are good candidates for expansion. For example, adjectives referring to countries have similar heads ("the Japanese/Chilean capital", "the Japanese/Chilean government"), but adding "Japanese" to a query that contains "Chilean" will rarely produce good results. Note that there are many words that distinguish "Japanese" and "Chilean" in terms of coocurrence in a sentence: "Tokyo", "Andes", "Samurai", etc. Grefenstette (1992) demonstrates that head-modifier-based term expansion can improve retrieval performance. However, we achieve performance improvements that are at least as good without parsing technology.

Qiu and Frei 1993 present an elegant and computationally efficient way of thesaurus construction. However, although the construction of the thesaurus is simple, the time complexity of computing the similarity between terms is linear in the size of the document collection, so that their scheme cannot be directly applied to a large collection like TIPSTER.

3 A Cooccurrence-Based Thesaurus

The basic idea of the thesaurus is to use a (symmetric) term-by-term matrix C, in which element c_{ij} records the number of times that words i and j cooccur in a window of a certain size (40 words in our experiment). Topical or semantic similarity between two words can then be computed as the cosine between the corresponding columns of C. The assumption is that words with similar meanings will occur with similar neighbors if enough text material is available. But this simple scheme cannot be implemented directly because we will consider 450,000 unique terms from the TIPSTER text collection (Harman 1993). A 450,000-by-450,000 matrix is unmanageably large for automatic computation.

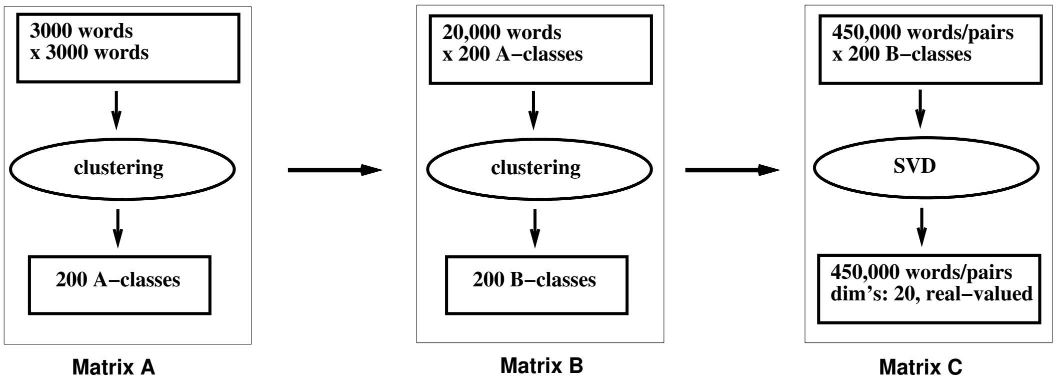

Figure 1: Iterative thesaurus construction.

Therefore, the thesaurus is constructed in three steps as shown in Figure 1. The final step is a singular value decomposition in which the dimensionality of "word space" is reduced in order to get more compact representations. Since SVD is expensive, $(O(n^2))$, the dimensionality of the original space cannot be too high. For this reason, the induction starts by deriving a set of two topical word classes from the corpus (the A-classes and B-classes in the figure) which then serve as the initial dimensions of word space.

The topical classes were derived in two iterations. First, a cooccurrence matrix A for 3,000 medium frequency words (frequency ranks 2,000 through 5,000) was collected (see "Matrix A" in Figure 1). Element $a_{i,j}$ of the matrix recorded the number of times that words w_i and w_j cooccured in a window of 40 words in the text collection. The columns of the matrix were then clustered into 200 A-classes $g_{A1}, g_{A2}, \ldots, g_{A200}$ using group average agglomerative clustering (Gnanadesikan 1977) and the cosine to measure similarity between columns.

A second cooccurrence matrix B was then collected for the 20,000 most frequent words, excluding stop words (see "Matrix B" in Figure 1). Each element $b_{i,j}$ recorded the number of times that word w_j coocurred with any of the medium-frequency words from class g_{Ai}. The 20,000 columns of the matrix were clustered into 200 B-classes $g_{B1}, g_{B2}, \ldots, g_{B200}$ using the fast clustering algorithm Buckshot (Cutting et al. 1992).[2] The purpose of the second iteration of the class-derivation procedure is to make sure that each word in the corpus has sufficiently many neighbors from at least one of the classes. With only 3,000 words representing the 200 classes, many words would not have any such neighbors.

Finally, a third cooccurence matrix C was collected for all 176,116 words that occurred at least twice in the collection and for all 272,914 pairs of adjacent words that occurred at least 5 times (see the right part of Figure 1). Element $c_{i,j}$ contained the number of times that word or pair j cooccurred in a window of 40 words with the subset of the 20,000 words in class g_{Bi}.

A SVD dimensionality reduction to 20 dimensions was used to improve generalization and compactness of the 200-component thesaurus vectors (Deerwester et al. 1990,Berry 1992). Since an application of SVD to the whole of C (a 200-by-449030 matrix) would be computationally too expensive, only a subset of the matrix, corresponding to the 1000th through 6000th most frequent word was decomposed. This decomposition defines a mapping from the 200 dimensional word-class space to a 20 dimensional reduced space. By applying the mapping to each of the 449,030 200-component class-based vectors, a smaller 20-dimensional vector was computed for each word and pair.

[2] A random sample of 2,000 of the 20,000 words was clustered first, and this clustering was then extended to all 20,000 words by assigning every word to the cluster whose centroid it was closest to.

The benefits of the singular value decomposition are higher speed in processing, since smaller vectors can be used, and improved generalization. Similarity between 200-component vectors is often not a good measure for semantic similarity since there may be word classes with similar topics. For example, class g_{B4} contains words like "navy", "radar", and "missile", and some of the member of class g_{B47} are "tanks", "missiles", and "helicopters". If one of two words has many neighbors in g_{B4} and the other in g_{B47} then they would not be similar in the 200-dimensional space, but they are similar in the reduced space. This is because the SVD algorithm recognizes and eliminates the redundancy.

4 An IR system with cooccurrence-based expansion

Figure 2 presents the architecture of our system. First, the documents of a text collection are preprocessed to induce the thesaurus. A query is then processed by expanding both the documents and the query with close terms from the thesaurus. A similarity search yields the top ranking documents with respect to the query. In what follows, we describe the vector expansion component and our results.

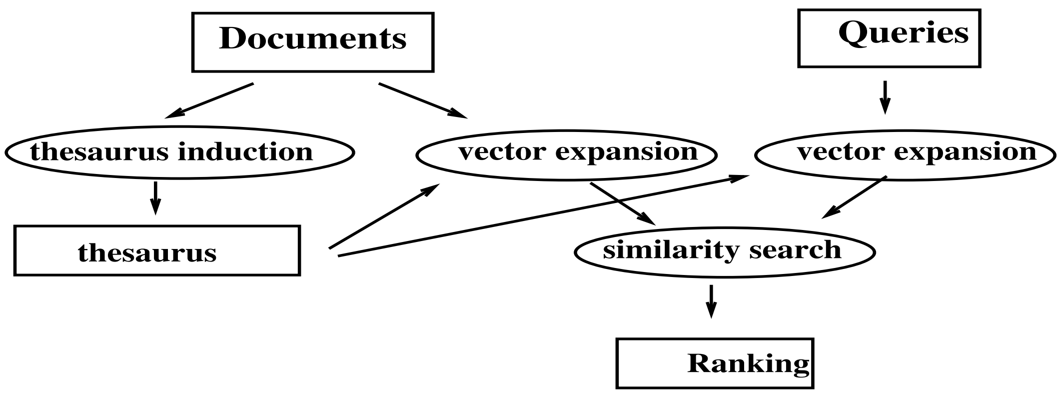

Figure 2: The system architecture.

In the basic vector similarity model, queries and documents are represented as vectors in term space. We used the augmented tf.idf scheme for weighting terms: (Salton and Buckley

min. freq. / # terms	7	8	10	12	15
1	0.2754				0.2745
2		0.2828	0.2817	0.2823	0.2833
3			0.2817		0.2714

Table 1: Average precision for different minimum frequencies and expansions per term.

1990)

$$w_{ij} = (0.5 + 0.5 * \frac{tf_{ij}}{max_i(tf_{ij})}) * log(\frac{N}{n_i})$$

where tf_{ij} is the frequency of word i in document j, N is the total number of documents, n_i is the document frequency of word i.

The only modification for vector expansion is the use of the expanded term frequency tf'_{ij} for unexpanded tf_{ij}:

$$tf'_{ij} = tf_{ij} + \sum_l tf_{lj}\delta_k(il)$$

where $\delta_k(il)$ is 1 if term l is one of the k nearest neighbors of term i and 0 otherwise. We use the cosine measure between vectors to measure closeness. The document frequencies n'_i are also recomputed to reflect changed document frequencies after expansion.

It turned out that retrieval performance decreased if we used all terms for expansion. We hypothesize that this is because little information is present in the cooccurrence patterns of rare words. Synonyms and near-synonymns of a word that occurs only two or three times are hard to identify from its cooccurrence patterns. Therefore, we limited expansion to terms that occurred at least m times, for different values of m. This frequency limit was applied both to expanded terms and expanding terms, that is, only terms with a frequency greater than m were added to the document representation and only terms with a frequency greater than m were expanded.

Table 1 shows average precision for several values of m (minimum frequency) and k (number of expanding terms per expanded term). The table suggests that the thesaurus does not provide sufficient information for expansion with more than 2 words. The thesaurus construction

method	average precision
no expansion	0.2762
minimum 15, 2 expanded	0.2833
rank combination	0.2989

Table 2: Tf/idf baseline and best performing vector expansion experiments.

algorithm is less reliable for low-frequency words as the mediocre result for $m = 7$ show (several experiments with higher values for k (number of expanding terms) lowered average precision even further). Performance for $m = 7$ is actually worse than the average precision without expansion (0.2762). Previous experiments with very rare words ($m < 7$) that were performed on the Cranfield collection showed that their expansion decreased performance.

The best result is achieved for $k = 2$ and $m = 15$ (see Table 2). This is only a small improvement over tf.idf. The retrieval algorithm was modified to further improve these results. The ranks from expanded and unexpanded retrieval were combined in the following way. Documents were ranked according to similarity with the query in expanded and unexpanded word space (ranks r_i^{exp} and r_i^{unexp} respectively) and the ranks were summed for each document. The final ranking was then the ranking of the summed ranks r'_i. Average precision of the combined ranking is 8% better than tf.idf in relative terms (2% in absolute terms, see Table 2).

$$r'_i = r_i^{\mathrm{exp}} + r_i^{\mathrm{unexp}}$$

Figure 3 shows precision for 11 recall points for unexpanded retrieval (bottom line) and for the combination of expanded and unexpanded retrieval.

5 Conclusion

We have shown that cooccurrence-based vector expansion can significantly improve information retrieval. In the future, we are planning to improve our thesaurus construction algorithm, with the goal of a closer correspondence between vector similarity in the thesaurus space and semantic similarity. We hope that an improved thesaurus would further enhance the performance

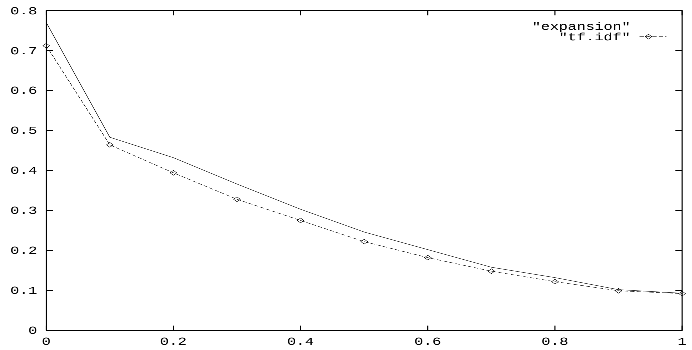

Figure 3: Precision for 11 recall points for combination of expansion and tf.idf (top line) and tf.idf (bottom line).

of information retrieval systems.

References

Berry, M. W. 1992. Large-scale sparse singular value computations. *The International Journal of Supercomputer Applications* 6(1):13–49.

Chute, C. G., Y. Yang, and D. A. Evans. n.d. Latent semantic indexing of medical diagnoses using umls semantic structures.

Crouch, C. J. 1990. An approach to the automatic construction of global thesauri. *Information Processing & Management* 26(5):629–640.

Cutting, D. R., J. O. Pedersen, D. Karger, and J. W. Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of SIGIR '92*, 318–329.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Evans, D. A., K. Ginther-Webster, M. Hart, R. G. Lefferts, and I. A. Monarch. 1991. Automatic indexing using selective nlp and first-order thesauri. In *Proceedings of the RIAO*, Vol. 2, 624–643.

Gnanadesikan, R. 1977. *Methods for Statistical Data Analysis of Multivariate Observations*. New York: John Wiley & Sons.

Grefenstette, G. 1992. Use of syntactic context to produce term association lists for text retrieval. In *Proceedings of SIGIR '92*, 89–97.

Harman, D. 1993. Overview of the first trec conference. In *Proceedings of SIGIR '93*.

Liddy, E. D., and W. Paik. 1992. Statistically-guided word sense disambiguation. In R. Goldman (Ed.), *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*. AAAI Press.

Peat, H. J., and P. Willett. 1991. 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.

Qiu, Y., and H. Frei. 1993. Concept based query expansion. In *Proceedings of SIGIR '93*.

Ruge, G. 1992. Experiments on linguistically-based term associations. *Information Processing & Management* 28(3):317–332.

Salton, G., and C. Buckley. 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science* 41(4):288–297.

Salton, G., and M. J. McGill. 1983. *Introduction to modern information retrieval*. New York: McGraw-Hill.

Schütze, H. 1992. Dimensions of meaning. In *Proceedings of Supercomputing '92*, 787–796.

Sparck Jones, K. 1971. *Automatic keyword classification for information retrieval*. London: Butterworth.

Voorhees, E. M., and Y.-W. Hou. 1992. Vector expansion in a large collection. In *Proceedings of TREC*.