Term Clustering of Syntactic Phrases

David D. Lewis, W. Bruce Croft
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 90-71
August 1990

*To appear in Proceedings of the Thirteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Brussels, 1990. The version in this report contains some typographic corrections.

Term Clustering of Syntactic Phrases

David D. Lewis
W. Bruce Croft
Computer and Information Science Department
University of Massachusetts, Amherst, MA 01003
To appear in SIGIR-90

April 30, 1990

Abstract

Term clustering and syntactic phrase formation are methods for transforming natural language text. Both have had only mixed success as strategies for improving the quality of text representations for document retrieval. Since the strengths of these methods are complementary, we have explored combining them to produce superior representations. In this paper we discuss our implementation of a syntactic phrase generator, as well as our preliminary experiments with producing phrase clusters. These experiments show small improvements in retrieval effectiveness resulting from the use of phrase clusters, but it is clear that corpora much larger than standard information retrieval test collections will be required to thoroughly evaluate the use of this technique.

1 Introduction

A primary goal of information retrieval (IR) research has been the development of methods for converting the original words of a document into a set of more effective content identifiers. Several of these representation methods make use of relationships between words in the original text. *Term clustering* attempts to group terms with related meanings, so that if any one appears in a query all can be matched in documents. *Syntactic phrase indexing* uses syntactic parsing to find groups of words in particular syntactic relationships, and indexes a document on these groups. Both of these methods have yielded mixed results in past experiments.

Reliably producing better representations requires understanding the important characteristics of representations and how they are change under different transformations [20]. Transformations applied to natural language text should take into account the fact that text contains more distinct words than is optimal for the statistical classification methods used in IR, and that as indexing terms these words are redundant, noisy, and infrequent. From a semantic standpoint, words are ambiguous identifiers of content, and are perhaps broader in meaning than is desirable.

Term clustering is a method which groups redundant terms, and this grouping reduces noise and increases frequency of assignment. If there are fewer clusters than there were original terms, then dimensionality is reduced as well. However, semantic properties suffer, since ambiguity can only be increased and meaning broadened.

Syntactic phrase indexing has exactly opposite effects. Each word in a phrase provides a context which disambiguates the other, and the meaning of a phrase is narrower than that of its component words. However, statistical properties suffer, since a large number of terms, many of them redundant and infrequently assigned, are created.

The strengths of these two methods are complimentary, which leads us in this paper to investigate combining them. We begin by surveying previous research in both areas. Following that, we discuss the specifics of our syntactic phrase generator and the phrases formed.

We then turn to the clustering of phrases. While the low frequency of occurrence of phrases makes them desirable to cluster, it also makes traditional similarity measures based on co-occurrence in documents untenable. We have instead formed clusters based on co-occurrence in semantically coherent groups of documents defined by controlled vocabulary indexing. These initial experiments produced only small performance improvements, and indicated that much larger corpuses will be necessary to produce high quality phrase clusters.

2 Previous Research

In this section we survey previous research on term clustering and syntactic indexing, as well as work near the intersection of the two areas. Our goal in this survey is to identify what has been learned about these techniques and how they might be combined.

2.1 Term Clustering

Term clustering is the application of cluster analysis [1] to forming groups of terms drawn from an existing text representation. From a pattern recognition viewpoint, term clustering is a form of *feature extraction*—a way of transforming an initial set of features into a new set that is more useful for classifying patterns (in this case, documents) [15]. It is therefore related to other feature extraction methods that have been used in IR, such as document clustering and factor analysis.

Any cluster analysis method requires that some similarity (or dissimilarity) measure be defined on the items to be clustered. Term clustering in IR has usually, though not always, defined similarity in terms of the degree to which two terms occur in the same documents.

Term clustering has been widely researched, with the largest body of work performed by Sparck Jones [35,32,36]. She investigated the effect of clustering strategies, term similarity measures, and vocabulary characteristics on the performance achieved with a clustered representation. Some of her most important conclusions were that clusters should be restricted to relatively infrequent and highly similar terms, clusters should be used to supplement the

original terms rather than replace them, and that clustering was unlikely to be effective if the relevant and non-relevant documents were not well separated on the input representation. The particular shape of clusters formed and the particular measure of similarity between terms was not found to have a significant effect. Of the several collections she experimented with, only one had its retrieval performance significantly improved by term clustering.

Similar early experiments were performed by Salton and Lesk [27], Lesk [18], and Minker, et al [23]. Salton and Lesk compared statistical term clustering with manually constructed thesauri on three test collections. No significant performance improvements were found for the term clustering, in comparison with significant improvements for two out of three collections for the manual thesauri.

Lesk's experiments were, strictly speaking, with association lists rather than clusters, the difference being that a term A can be considered similar to a term B without the reverse holding. Lesk expanded both query and document descriptions with similar terms of moderate collection frequency, but achieved no large performance improvements. Lesk studied the term similarities that were actually produced and concluded that the small size of his collections (40,000 to 110,000 words) meant that the similarities were local to the collections, and were not good indications of the general meanings of the words.

Minker and colleagues experimented with two collections, and with three different text representations for each. Terms from all six representations were clustered using a variety of graph-theoretic algorithms. Like Sparck Jones, Minker found that small clusters performed the best, but he found no significant performance improvements over indexing on terms.

All of the above researchers used co-occurrence in documents as the basis for term similarity. Other similarity measures include co-occurrence in syntactic relationships with particular words [14] and presence in pairings between queries and relevant documents [40]. Crouch recently achieved significant performance improvements on two collections by first clustering documents, and then grouping low frequency terms that occurred in all documents of a document cluster [7].

2.2 Research on Syntactic Phrase Indexing

The use of syntactic information for phrasal indexing has been surveyed elsewhere [9,31,21], so we discuss this area only briefly. These techniques break down into two major classes: template-based and parser-based.

Dillon and Gray's FASIT system [8] is typical of template-based phrasal indexers. Adjacent groups of words from documents are matched against a library of templates, such as <JJ-NN NN> (adjective noun), and <NN PP NN> (noun preposition noun), and those matching some template are retained. Most templates in FASIT and other template-based systems are oriented toward finding contiguous words which represent noun phrases. Phrases are normalized by stemming and removal of function words. Klingbiel's MAI system used a similar strategy [16], while the TMC Indexer [24] and LEADER [13] combined limited parsing with templates.

Parser-based strategies attempt to analyze entire sentences or significant parts of them in

producing syntactic phrases. Fagan [9], for example, used the PLNLP parser to completely parse the text of two test collections and extract indexing phrases. The sophistication of the PLNLP grammar enabled Fagan to handle complex noun phrases with prepositional and clausal postmodifiers, as well as some adjectival constructions. Fagan also used a number of hand-built exclusion lists of words which signaled that a phrase should not be generated or should be generated in a special fashion.

On two test collections Fagan's syntactic phrases produced improvements of 1.2% and 8.7% over indexing on words alone. Despite the care with which Fagan's phrases were formed, this was less than the improvement (2.2% and 22.7%) provided by very simple statistically defined phrases. Furthermore, Sembok's system [30] achieved similar results to Fagan using only a very simple noun phrase grammar. Smeaton's method [31] provided a somewhat smaller improvement over single word indexing than the above two systems, but required parsing only of noun phrases in queries, followed by looking for co-occurrence of phrase components in documents.

In summary, experiments on syntactic phrase formation have not found it superior to statistical phrase formation, and have not found much correlation between the sophistication of phrase formation and the resulting performance improvements.

2.3 Integration of Syntactic Phrase Indexing and Clustering

While there has been extensive research on both term clustering and syntactic phrase indexing, the two techniques have not been directly combined before. Of course, almost all phrase generation systems in effect do a small amount of clustering when they normalize phrases, mainly through stemming. The FASIT system combined all phrases which had a particular word in common into a group, a very simple form of clustering which did not appear to be very effective. Antoniadis, et al describe a similar method, but it is not clear if it was actually used in their system [2].

More traditional statistical clustering techniques have been used in at least two IR interfaces to suggest terms, including syntactically formed phrasal terms, that a user might want to include in their query. The LEADER system formed cliques of phrases based on co-occurrence in full document texts, and the REALIST system used unspecified statistical techniques to provide lists of strongly correlated terms [37]. Neither study presented any performance data resulting from the use of these strategies, however.

Salton [28] investigated indexing documents on *criterion trees*. These were equivalent to hand-constructed clusters of syntactic structures, with individual words replaced by class labels from a manually constructed thesaurus. A related strategy is Sparck Jones and Tait's [34] generation of groups of alternative indexing phrases from a semantic interpretation of a query. The phrases generated by this method only contained words from the query, but a thesaurus could have been used, as with criterion trees. Neither of these methods were tested on large enough collections to draw firm conclusions about their efficacy, and neither thoroughly addresses the statistical problems with syntactic phrases.

Lochbaum and Streeter have recently reported on the use of a factor analysis technique, singular value decomposition (SVD), to compress term-document matrices [22]. They found

that the inclusion of some noun phrases in addition to single words improved the performance achieved with the compressed representation. Since SVD can be viewed as simultaneously performing a term clustering and a document clustering, this result is suggestive that term clustering of phrases will provide an improved representation.

SVD can take advantage of dependencies among both terms and documents, but has the disadvantage that it is currently too computationally expensive for use with large document collections. Another advantage of term clustering over SVD is that prior knowledge about likely term groupings is more easily incorporated into a clustering similarity function than into the term-document matrix.

2.4 Summary

Previous research in the areas of term clustering and syntactic indexing has revealed a few important guidelines, and considerable evidence that many other choices are not of much significance. With respect to term clustering, the particular clustering algorithm used has not been found to make much difference, as long as clusters are small and composed of low frequency terms.

There is some evidence that the method by which the similarity of terms is judged can have an important effect. Crouch's strategy, for instance, partially addresses the dilemma that infrequent terms are the ones that most benefit from clustering, but are also the most difficult to get accurate co-occurrence data on.

Probably the most surprising result of research on syntactic phrase indexing is that the linguistic sophistication of the phrase generation process appears to have little effect on the performance of the resulting phrase representation. An open question is whether parser-based approaches are superior to template-based ones, but at least so far both approaches have been found inferior to statistical phrases.

Since it seems unlikely that individual statistical phrases are better content indicators than individual syntactic phrases, this suggests that it is the poor statistical properties of syntactic phrases (high dimensionality, noise, etc.) that are at fault. Clustering of syntactic phrases is a natural approach to improving these properties. While there is no direct evidence available on the performance of syntactic phrase clustering, the SVD results are encouraging.

3 Extracting Syntactic Phrases

This section first describes a particular goal for phrase formation and how our system approximated this ideal. We then show some of the strengths and weaknesses of the system by the analysis of an example sentence. Finally, we present statistics on phrase formation for the CACM-3204 corpus.

3.1 Syntactic Analysis Technology

One factor that makes previous research on syntactic indexing hard to evaluate is the wide range of heuristic techniques used in generating syntactic phrases. Since none of these variations has proven strikingly superior to others, we opted for a definition of phrases which was as simple as possible linguistically. We defined a syntactic phrase to be any pair of non-function words in a sentence that were heads of syntactic structures connected by a grammatical relation. Examples are a verb and the head noun of noun phrase which is its subject, a noun and a modifying adjective, a noun and the head noun of a modifying prepositional phrase, and so on. This is essentially the definition used by Fagan [9], except that we form phrases from all verbal, adverbial, and adjectival constructions, and do not maintain exclusion lists of specially treated words.

It is important to distinguish the definition of syntactic phrases used by a system from the actual set of phrases produced. Current syntactic analysis systems are far from perfect, so any definition of syntactic phrases which is not of the form "syntactic phrases are what my program produces" can only be approximated. Even the PLNLP parser used by Fagan produced a correct analysis of only 32% of a typical set of sentences [29], and that system was the result of a large-scale corporate development effort.

In designing our phrase generation system we attempted to generate all phrases that suited our definition, while avoiding the complexity and ambiguity of producing a full parse for each sentence. Our approach was to parse only the constituents of a sentence below the clause level. The analysis of a sentence, therefore, was a sequence of noun phrases, adjective phrases, adverb phrases, verb groups, and miscellaneous punctuation and function words. Since much of the complexity of most grammars is in rules to capture clause level structure, we were able to restrict ourselves to a grammar of only 66 rules.

Limiting the complexity of analysis does not limit the need for a large lexicon, since every word still had to be interpreted. We used the machine-readable version of the Longman Dictionary of Contemporary English (LDOCE) [3], which provided syntactic categories for about 35,000 words. By using a morphological analyzer for inflectional suffixes we extended the effective vocabulary of the system to perhaps 100,000 words. Even so, a substantial number of words encountered in text were not present in the dictionary. These tended to be compound words, proper nouns, or very technical terms. These unknown words were assumed to be ambiguous between the categories **noun**, **verb**, and **adverb**, and were allowed to be disambiguated by the grammar.

Parsing was performed by a chart parser operating in bottom-up mode[1]. The bottom-up parsing strategy produced a large number of overlapping parse trees covering parts of the sentence. The parser then selected a small set of non-overlapping trees which together covered the entire sentence. Phrase formation used these trees in two ways. Phrases were generated from complete constituents by means of annotations to each grammar rule. These annotations indicated which components of a tree corresponding to that rule should be combined into a phrase.

[1]The parser was designed and implemented by John Brolio at the University of Massachusetts, who also was the principal designer of the syntactic grammar.

It sometimes was desirable to produce phrases from neighboring constituents as well. For instance, if a verb group was followed by a noun phrase, we wanted to combine the verb with the head noun of the noun phrase. Heuristics for forming phrases under these circumstances, including the handling of conjunction, punctuation, and function words, were encoded in a small (5 state) pushdown automaton.

Note that the two words in a phrase were considered to be unordered, and no distinction was made between phrases formed from different syntactic structures.

3.2 An Example of Phrase Generation

As an example, consider the following sentence from the CACM-3204 collection:

> *Analytical, simulation, and statistical performance evaluation tools are employed to investigate the feasibility of a dynamic response time monitor that is capable of providing comparative response time information for users wishing to process various computing applications at some network computing node.*

A complete and correct analysis of this sentence would be extremely complex and would have to be distinguished from a large number of plausible alternatives. However, the partial syntactic constituents produced by our system capture most of the structure necessary to produce reasonable phrases. The greatest advantage of this approach is that reasonable analyses can be produced for any sentence. In Figure 1 we show the phrases that would be produced from a perfect parse of the sentence, and those that were produced by our system. Bracketed phrases are ones that would not have been produced by a perfect system, though some are reasonable indexing phrases.

The phrases generated from this example sentence exhibit some of the strengths and weaknesses of our system. For instance, the words *analytical, statistical, evaluation,* and *feasibility* were not present in the lexicon. Grammatical constraints were able to disambiguate *evaluation* and *feasibility* correctly to nouns, while *analytical* and *statistical* were incorrectly disambiguated to nouns. However, the incorrect disambiguations did not affect the generation of phrases, since premodifying nouns and adjectives are treated identically.

The presence of a word in LDOCE did not guarantee that the correct syntactic class would be assigned to it. The words *tool, dynamic, time, monitor, provide/providing, comparative, wish* and *process* all had multiple syntactic classes in LDOCE. Of these, *dynamic, providing, comparative, wishing,* and *process* were disambiguated incorrectly. The only case where phrase generation was seriously interfered with was in the interpretation of *providing* as a conjunction.[2] This meant that the phrases *providing information* and *providing users* were not generated. The interpretation of *wishing* and *process* as nouns, and the resulting interpretation of a clausal structure as a noun phrase, while atrocious from a linguistic point of view, had a relatively minor effect on phrase generation.

[2] One price of using a machine-readable dictionary as a syntactic lexicon is the occasional odd classification.

DESIRED PHRASES	PHRASES PRODUCED
analytical tools	*<analytical employed>*
simulation tools	*<employed simulation>*
statistical tools	*statistical tools*
performance evaluation	*<performance tools>*
evaluation tools	*evaluation tools*
tools employed	*tools employed*
employed investigate	*employed investigate*
investigate feasibility	*investigate feasibility*
feasibility monitor	*feasibility monitor*
response time	*<response monitor>*
time monitor	*time monitor*
dynamic time	*<dynamic monitor>*
monitor capable	
capable providing	*<capable feasibility>*
providing information	*<capable information>*
comparative information	*comparative information*
response time	*<response information>*
time information	*time information*
information users	*<information wishing>*
	<information applications>
users wishing	*users wishing*
wishing process	*<wishing applications>*
process applications	*process applications*
various applications	*various applications*
computing applications	*computing applications*
process node	*<applications node>*
	<wishing node>
network node	*network node*
computing node	*computing node*

Figure 1: Desired and Actual Phrases (Before Stemming) for Example Sentence.

Collection Frequency (in 1425 Docs)	Unstemmed		Stemmed	
	Number of Distinct Phrases	Total Phrase Occurrences	Number of Distinct Phrases	Total Phrase Occurrences
1	41500	43612	32470	34689
2	3399	7336	4056	8866
3	906	3015	1284	4299
4	370	1687	576	2584
5	169	963	309	1735
6	124	850	218	1503
7	57	443	108	855
8	47	458	90	814
9+	128	2157	281	5176
Total	46700	60521	39392	60521

Table 1: Statistics on Phrase Generation for 1425 CACM Documents

3.3 Phrase Statistics

For the experiments reported in this paper we parsed and generated phrases from the titles and abstracts of 1425 documents, totaling 110,198 words, from the CACM-3204 collection. We used only those documents which have *Computing Reviews* categories assigned to them, since our current clustering strategy requires that controlled vocabulary indexing be available for documents. Table 1 breaks down the phrases generated according to the number of times they occurred in these 1425 documents.

As expected, the number of phrases was very large, and relatively few phrases had many occurrences. We used the Porter stemmer [26] to stem the words in phrases, which increased phrase frequency somewhat. These stemmed phrases were used for all the experiments reported in this paper.

4 Clustering Phrases

Given the few differences found between text representations produced by different clustering algorithms, we chose to form the very simple clusters that Sparck Jones referred to as *stars* [32]. These clusters consist of a seed item and those items most similar to it. A fixed number of nearest neighbors, a minimum similarity threshold, or both can be used. Here are some randomly chosen example clusters formed from CACM phrases when clusters were restricted to a size of 4:

{ <*linear function*>, <*comput measur*>, <*produc result*>, <*log bound*> }
{ <*princip featur*>, <*draw design*>, <*draw display*>, <*basi spline*>, <*system repres*> }
{ <*error rule*>, <*explain techniqu*>, <*program involv*>, <*key data*> }
{ <*substant increas*>, <*time respect*>, <*increase program*>, <*respect program*>}

The seed phrases are underlined above. Some clusters contain more than 4 elements, since elements with negligibly greater dissimilarity to the seed than the fourth element were also retained.

The clusters formed rarely contained any exact synonyms for the seed phrase. This is not surprising since, of the large number of phrases with a given meaning, one will usually be considerably more frequent than the others. Given the relatively small size of the CACM corpus, only the most frequent of the synonymous phrases will have more than one occurrence. Since we required that a phrase must occur at least in at least two documents to be clustered, synonymous phrases were almost never clustered. However, some good clusters of closely related phrases were formed, along with many accidental clusters of essentially unrelated phrases.

The rest of this section discusses how clusters were formed and how they were used in scoring documents. Section 5 will then discuss our experimental results.

4.1 Co-occurrence In Controlled Vocabulary Indexing Categories

The dilemma between the desire to cluster infrequent terms and the lack of information on which to judge their similarity is even more severe for phrases than for words. Given that only 1.8% of the distinct phrases in our corpus occurred more than 5 times, it was unreasonable to expect that many phrases would have any substantial number of co-occurrences in documents.

Crouch's strategy of looking for co-occurrence in document clusters was a promising alternative, but we were conscious of the fact that document clustering itself does not necessarily produce meaningful clusters. Therefore, instead of producing document clusters, we made use of the document clustering implicit in the controlled vocabulary indexing of the CACM collection. A total of 1425 of the CACM documents, are indexed with respect to a set of 201 *Computing Reviews* (*CR*) categories [11,19]. Of those categories, 193 are assigned to one or more documents. Since *CR* categories are arranged in a three-level hierarchy, we assumed that whenever a document was assigned to a category it was also assigned to all ancestors of that category.

Some method was then required for clustering the phrases based on their presence in the *CR* categories. Crouch found the set of low frequency terms in each of the documents in a cluster and took the intersection of these sets. The large and quite variable size of the *CR* clusters makes this strategy inappropriate for us. Instead we viewed each *CR* category as a feature on which a phrase could take on a value between 0 and 1. We used the value n_{pc}/n_c, where n_{pc} was the number of occurrences of phrase p in category c, and n_c was the total number of occurrences of all phrases in category c. This treated multiple occurrences of a phrase as being more significant than single occurrences, and also normalized for the large differences in the number of documents, and thus phrases, appearing in the different categories.

The cosine correlation was used to compute the similarity between feature vectors for different phrases. This had the effect of normalizing for overall phrase frequency. All phrases occurring in 2 or more documents were used in clustering, expect when otherwise mentioned in results.

4.2 Weighting of Clusters

The point of forming clusters, of course, was to use them in retrieval. This required a method for incrementing the scores of documents based on the presence of phrases and clusters of phrases in queries and documents. We chose to use the same weighting methods used by Fagan for phrases and by Crouch for clusters, since these methods have shown some effectiveness in the past.

Fagan [9,10] assigned a two-word phrase a weight (in both queries and documents) equal to the mean of the weights of its component stems. The stem weights themselves are computed as usual for the vector space model. The inner products were computed separately for terms and phrases and then added together, potentially with different weightings.

Crouch [7] used a very similar method for clusters, giving them a weight in a query (or a document) equal to the mean of the weights of the cluster members in the query (or the document). The resulting weights were then multiplied by 0.5 in both documents and queries, for an overall downweighting factor of 0.25 for clusters with respect to single terms.

Combining these gave the following similarity function to be used for ranking documents:

$$SIM(q,d) = (c_s \cdot ip(q_s, d_s)) + (c_p \cdot ip(q_p, d_p)) + (c_c \cdot ip(q_c, d_c))$$

where ip is the inner product function, q_s, q_p, and q_c are the weight vectors of stems, phrases, and phrase clusters for queries, d_s, d_p, and d_c are the vectors for documents, and c_s, c_p, and c_c are the relative weights of stems, phrases, and documents.

5 Experiments

The main goal of the experiments reported here was to discover whether applying clustering to phrases from a small corpus would result in an improved text representation. Another goal was to explore whether the factors which have been found to be most important in clustering of words also have a strong impact on clustering of phrases. These include the size of clusters formed, the frequency of items clustered, and the maximum dissimilarity tolerated between cluster members. A secondary goal was to gather preliminary data on the efficiency of syntactic phrase clustering, given the likelihood than larger scale clustering would have to be investigated. We report on each of these goals in the following sections.

All retrieval results are based on the full CACM collection of 3204 documents. We used only the 50 queries which do not request documents by particular authors, and for which there are one or more relevant documents.

Recall Level	Precision					
	Clusters + Terms				Phrases	
	Size 2	Size 4	Size 8	Size 12	+ Terms	Terms
0.10	55.5	55.5	57.9	57.1	58.1	56.3
0.20	43.2	42.0	42.2	41.9	45.4	41.0
0.30	37.7	37.0	36.5	36.2	38.0	35.7
0.40	31.1	30.5	30.8	30.0	30.2	29.6
0.50	23.3	23.3	22.2	22.3	23.4	22.0
0.60	19.5	19.3	18.2	18.3	19.0	18.8
0.70	13.5	13.3	13.3	13.3	13.7	13.8
0.80	9.2	9.4	9.4	9.3	9.5	9.9
0.90	5.5	5.8	5.6	5.6	5.6	6.1
1.00	4.2	4.1	4.1	4.1	4.1	4.7
Avg. Prec.	24.3	24.0	24.0	23.8	24.7	23.8
Change	+2.1%	+0.8%	+0.8%	+0.0%	+3.8%	

Table 2: Performance Using Clusters and Terms

5.1 Effectiveness of Syntactic Phrase Clusters

Our first concern was whether the clusters of syntactic phrases formed from this small corpus would be sufficient to improve retrieval performance. Table 2 compares recall and precision figures for 4 sizes of clusters to the figures for single terms (stems) and single terms combined with syntactic phrases. Clusters produce a slightly smaller improvement than phrases, and neither is significantly better than the use of single terms alone.

Using both clusters and phrases (Table 3) provides the most improvement. These results would be classified as "noticeable" ($> 5.0\%$) but not "material" ($> 10.0\%$) according to Sparck Jones' criteria [33]. We investigating varying the weighting of the cluster and phrase vectors (c_c and c_p, respectively), but found only trivial and inconsistent improvements resulting from any values besides 1.0. In particular, reducing weighting of clusters to Crouch's value of 0.25 caused a small decrement in performance, providing some evidence that clusters of phrases are better content indicators than clusters of words.

5.2 Factors Affecting Phrase Clustering

In our survey on term clustering, we mentioned a number of factors that had been found in the past to impact the effectiveness of term clustering. We have already mentioned the effect of cluster size. Sparck Jones found small, tight clusters, of size 2 to 4, to be most effective, and our results are in agreement with this. We also found that using clusters of phrases in addition to phrases, rather than instead of phrases, was most effective. This again is in agreement with Sparck Jones' results on clustering of single terms.

Another approach to forming tight clusters would be to require that phrases have no greater than a fixed dissimilarity with the seed phrase. This causes some phrases not to

Recall Level	Precision					
	Clusters + Phrases + Terms				Phrases + Terms	Terms
	Size 2	Size 4	Size 8	Size 12		
0.10	57.4	60.0	59.3	58.5	58.5	56.3
0.20	46.4	46.4	46.1	45.0	45.4	41.0
0.30	38.8	39.5	38.9	37.7	38.0	35.7
0.40	31.3	31.1	31.1	30.8	30.2	29.6
0.50	23.0	23.1	23.1	23.1	23.4	22.0
0.60	19.3	19.5	19.5	19.5	19.0	18.8
0.70	13.9	13.9	13.8	13.7	13.7	13.8
0.80	9.6	9.8	9.7	9.6	9.5	9.9
0.90	5.7	5.7	5.7	5.7	5.6	6.1
1.00	4.2	4.2	4.2	4.2	4.1	4.7
Avg. Prec.	25.0	25.3	25.1	24.8	24.7	23.8
Change	+5.0%	+6.3%	+5.5%	+4.2%	+3.8%	

Table 3: Performance Using Clusters, Phrases, and Terms

cluster at all. We investigated several dissimilarity thresholds for cluster membership, but found only trivial improvements, and some degradations, in performance.

Another factor which has been found to impact term clustering is the frequency of the terms being clustered. The exclusion of high frequency terms from clusters was found by Sparck Jones in particular to be important in achieving an effective term clustering. Maximum frequency thresholds used by Sparck Jones included 20 out of 200 (10%) documents, 20 out of 541 documents (3.6%), and 25 out 797 documents (3.1%) [36].

Since only 8 stemmed phrases occurred in more than 45 (3.2%) of the 1425 documents used for clustering, it was questionable whether omitting frequent phrases would be useful. We experimented with forbidding phrases which occurred in more than 45 documents from participating in clusters, and found this actually produced a slight decrease in performance. Forbidding phrases occurring in more than 30 documents produced a larger decrease. Examining the 8 phrases of frequency greater than 45 shows that even here there are several which are moderately good content indicators (*<oper system>*, *<comput program>*, *<program languag>*, *<comput system>*, *<system design>*) as well as several fairly bad ones (*<paper describ>*, *<paper present>*, *and <present algorithm>*). Therefore, omitting the most frequent phrases does not appear to be an appropriate strategy when clustering phrases.

One can also argue that very infrequent phrases should be omitted from clusters. If a term does not occur a sufficient number of times then we will have not have enough data on its distribution to accurately cluster it. Most work on term clustering has required that terms occur in 2 or more documents to become part of a cluster, but higher thresholds conceivably could result in more accurate clusters.

We investigated requiring that phrases occur in at least 3, 4, 5, or 6 documents in

order to be clustered. These were fairly severe restrictions considering the low frequency of phrases, resulting in reducing the number of phrases available for clustering from 6922 to 2866, 1582, 1015, and 706 respectively. Small performance improvements resulted for some of these restrictions in combination with some cluster sizes. However, the improvements vanished when clusters were used in combination with phrases as well as terms. These results do help confirm that the small amount of frequency data available on phrases was a major impediment to forming effective clusters.

5.3 Efficiency

Our results suggest that the use of corpuses much larger than CACM-3204 will be necessary if phrase clustering is to be an effective technique. This means that efficiency of clustering will be of considerable importance. We therefore conducted some preliminary investigations into efficiency methods.

The use of an inverted file to speed up the finding of nearest neighbors is a technique that has been applied to both document clustering [5,39] and term clustering [25]. The main advantage cited for this technique is the avoidance of calculating the large number of similarity values of 0 present in typical term-term or document-document matrices. These 0 values arise in term-term matrices when similarity is based on co-occurrence in documents, since most pairs of terms will not occur together in any document.

The term-term (i.e. phrase-phrase) similarity matrix in our experiments has few 0 values since some of the *CR* categories contain very large numbers of phrases. Most of the similarity values will be very small, however, since normalization by category size, in combination with the cosine similarity measure, ensures that co-occurring in large categories has only a small impact on similarity.

This normalization for category size means that the k nearest neighbors of a seed phrase will almost always share some relatively specific *CR* category with the seed. We can therefore adapt the technique, first proposed for document ranking [4], of searching inverted lists in order of their length and testing at the end of each list whether any unseen item can possibly be more similar than the k best items already seen. Using this technique we found that only 7.2% of phrases needed to be examined on average when forming clusters of size 2, which is similar to the reductions achieved when term-term matrices contain mostly 0's. Even so, a full clustering run took about 40 hours on a Texas Instruments Microexplorer workstation, so additional attention to clustering efficiency will clearly be necessary for larger corpora.

6 Analysis and Future Work

The small performance benefits reported above are disappointing, but not really surprising. The fact that a high proportion of the occurrences of phrases were of phrases which occur only once or twice means that a corpus on the order of 100,000 words is simply inadequate for producing phrase clusters. We have experiments underway on a corpus of over 1 million words of newswire text previously used in tests of a text classification system [12]. We have

also obtained corpora of 100 million words and more for future work.

Since standard IR test collections of large size are not currently available, the effectiveness of phrase clusters may have to be evaluated for retrieving documents which were not themselves used in forming the clusters. Previous researchers have suggested that the regularities captured by term clustering are collection dependent [35,18], which would interfere with this strategy. However, the combination of decreased ambiguity of phrases in comparison to words, combined with the use of a very large corpus, will, we believe, make phrase clusters of more general applicability.

To the extent that the CACM corpus allowed us to study the properties of phrase clustering, we found it to behave for the most part like clustering of single terms. The most notable exception was that excluding even the highest frequency phrases led to a degradation in performance. One possible explanation is that the corpus used is too small to manifest the frequency differences that would allow low quality phrases to be excluded. It should also be noted, however, that most of the results which argue against the clustering of high frequency terms assume ranking by coordination level. The use of inverse document frequency weighting may make exclusion of high frequency terms less important. A final possibility is that collection frequency is not as good an indicator of quality for phrases as it is for single terms. This view is supported by the fact that Fagan [9] found only trivial improvements in retrieval performance were possible from excluding high frequency syntactic phrases.

The exclusion of low quality phrases is clearly an important issue both for phrasal indexing and clustering of phrases. The fact that our performance improvements for syntactic phrases on the CACM collection are less than Fagan's (3.8% vs. 8.7%) suggests that his list of over 250 low content adverbs, verbs, and nouns, which triggered special purpose phrase generation heuristics, were successful in increasing the quality of phrases generated.

Some words which should be excluded from phrases can be defined linguistically, such as partitives (e.g. *half* in *eliminate half of the documents*). But many other words should be excluded from some corpuses and not from others. For instance, Fagan sensibly excluded the words "case," "property," and "development" from phrases for the computer and information science test collections he worked with, but this would not be appropriate in a collection of articles on real estate law. Word sense disambiguation methods might be useful in avoiding this problem [17].

The same distributional information used for clustering might also be usable to identify low quality phrases. In our experiments we noticed a tendency for low quality, high frequency phrases to appear under many different manual indexing categories, while high quality, high frequency phrases had most of their occurrences in a few categories. For instance, the low quality phrase <*paper describ*> occurs in 57 documents with a total of 104 *CR* categories assigned, while the higher quality phrase <*oper system*> occurs in 59 documents but only under 78 categories. Of course, as with clustering, a large text corpus is needed to obtain this distributional information.

Another potential source of high quality phrases is the user of the IR system [6]. While the user cannot control which phrases take part in clusters, he or she can control which

Recall Level	Precision					
	Clusters + Phrases + Terms				Phrases + Terms	Terms
	Size 2	Size 4	Size 8	Size 12		
0.10	60.7	61.9	61.5	61.4	61.4	56.3
0.20	45.8	45.9	45.9	45.9	45.2	41.0
0.30	40.6	40.3	39.8	39.8	39.5	35.7
0.40	34.2	33.4	33.5	33.5	33.2	29.6
0.50	25.0	25.1	25.2	25.2	25.3	22.0
0.60	19.8	20.7	20.7	20.6	20.9	18.8
0.70	13.8	14.6	14.5	14.6	14.6	13.8
0.80	9.4	10.2	10.0	10.0	10.0	9.9
0.90	5.6	6.3	6.2	6.3	6.2	6.1
1.00	4.2	4.9	4.9	4.9	4.9	4.7
Avg. Prec.	25.9	26.3	26.2	26.2	26.1	23.8
Change	+8.8%	+10.5%	+10.1%	+10.1%	+9.7%	

Table 4: Performance With Human-Selected Query Phrases

phrases are extracted from the query, and thus used to match clusters. If we restrict the phrases used in the CACM queries to ones identified by a human as meaningful[3], the performance of phrases and clusters increases considerably (Table 4).

Besides better methods for generating phrases and clusters of phrases, there is also a need for a better understanding of how to use them. The lack of theoretical underpinnings to heuristic weighting schemes such as Fagan's for phrases and Crouch's for clusters make it hard to have confidence that they will be effective on new collections. On the other hand, existing probabilistic retrieval models are inadequate for use with phrases and clusters, particularly in handling the known dependencies between terms and phrases and terms and clusters. Network models [38] and probabilistic models incorporating explicit dependencies are two promising alternatives [6].

7 Conclusions

Term clustering is a natural approach to remedying the poor statistical properties of syntactic phrases. Our preliminary experiments offer some encouragement that the technique is practical, though it is clear that much larger corpuses will be necessary to draw strong conclusions about the technique's potential to improve retrieval performance. A better understanding is also needed of methods for selecting appropriate phrasal identifiers, and of scoring documents based on phrase and cluster matches.

[3]The set of phrases used was generated by a graduate student who was not involved in the experiments on syntactic phrase formation.

Acknowledgments

We thank Longman Group, Ltd. for making available to us the typesetting tape for LDOCE, in the formatted version produced by Bran Boguraev at the University of Cambridge. This research was supported by the NSF under grant IRI-8814790, by AFOSR under grant AFOSR-90-0110, and by an NSF Graduate Fellowship. Anil Jain, Mike Sutherland, and Mel Janowicz provided advice on cluster analysis, and the work of our collaborator on syntactic parsing, John Brolio, was invaluable. Raj Das generated the hand-selected phrases for the CACM queries. All responsibility for errors remains with the authors.

References

[1] Michael R. Anderberg. *Cluster Analysis for Applications.* Academic Press, New York, 1973.

[2] Georges Antoniadis, Geneviève Lallich-Boidin, Yolla Polity, and Jacques Rouault. A French text recognition model for information retrieval system. In *Eleventh International Conference on Research & Development in Information Retrieval*, pages 67–84, 1988.

[3] Bran Boguraev and Ted Briscoe. Large lexicons for natural language processing: Utilising the grammar coding system of LDOCE. *Computational Linguistics*, 13(3–4):203–218, 1987. Special Issue on the Lexicon.

[4] C. Buckley and A. F. Lewit. Optimization of inverted vector searches. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 97–110, 1985.

[5] W. Bruce Croft. Clustering large files of documents using the single-link method. *Journal of the American Society for Information Science*, pages 341–344, November 1977.

[6] W. Bruce Croft and Raj Das. Experiments with query acquisition and use in document retrieval systems. In *Thirteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1990.

[7] Carolyn J. Crouch. A cluster-based approach to thesaurus construction. In *Eleventh International Conference on Research & Development in Information Retrieval*, pages 309–320, 1988.

[8] Martin Dillon and Ann S. Gray. FASIT: A fully automatic syntactically based indexing system. *Journal of the American Society for Information Science*, 34(2):99–108, March 1983.

[9] Joel L. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods.* PhD thesis, Department of Computer Science, Cornell University, September 1987.

[10] Joel L. Fagan. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, 40(2):115–132, 1989.

[11] Edward A. Fox, Gary L. Nunn, and Whay C. Lee. Coefficients for combining concept classes in a collection. In *Eleventh International Conference on Research & Development in Information Retrieval*, pages 291–307, 1988.

[12] Philip J. Hayes, Laura E. Knecht, and Monica J. Cellio. A news story categorization system. In *Second Conference on Applied Natural Language Processing*, pages 9–17, 1988.

[13] Donald J. Hillman and Andrew J. Kasarda. The LEADER retrieval system. In *AFIPS Proceedings 34*, pages 447–455, 1969.

[14] Lynette Hirschman, Ralph Grishman, and Naomi Sager. Grammatically-based automatic word class formation. *Information Processing and Management*, 11:39–57, 1975.

[15] J. Kittler. Feature selection and extraction. In Tzay Y. Young and King-Sun Fu, editors, *Handbook of Pattern Recognition and Image Processing*, pages 59–83. Academic Press, Orlando, 1986.

[16] Paul H. Klingbiel. Machine-aided indexing of technical literature. *Information Storage and Retrieval*, 9:79–84, 1973.

[17] Robert Krovetz and W. Bruce Croft. Word sense discrimination using machine-readable dictionaries. In *Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 127–136, 1989.

[18] M. E. Lesk. Word-word associations in document retrieval systems. *American Documentation*, pages 27–38, January 1969.

[19] David D. Lewis. A description of CACM-3204-ML1, a test collection for information retrieval and machine learning. Information Retrieval Laboratory Memo 90-1, Computer and Information Science Department, University of Massachusetts at Amherst, 1990.

[20] David D. Lewis. *Representation and Learning in Information Retrieval.* PhD thesis, University of Massachusetts at Amherst, 1990. In preparation.

[21] David D. Lewis, W. Bruce Croft, and Nehru Bhandaru. Language-oriented information retrieval. *International Journal of Intelligent Systems*, 4(3):285–318, 1989.

[22] Karen E. Lochbaum and Lynn A. Streeter. Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. *Information Processing and Management*, 25(6):665–676, 1989.

[23] Jack Minker, Gerald A. Wilson, and Barbara H. Zimmerman. An evaluation of query expansion by the addition of clustered terms for a document retrieval system. *Information Storage and Retrieval*, 8:329–348, 1972.

[24] Paul M. Mott, David L. Waltz, Howard L. Resnikoff, and George G. Robertson. Automatic indexing of text. Technical Report 86-1, Thinking Machines Corporation, January 1986.

[25] T. Noreault and R. Chatham. A procedure for the estimation of term similarity coefficients. *Information Technology*, pages 189–196, 1982.

[26] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[27] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the Association for Computing Machinery*, 15(1):8–36, 1968.

[28] Gerard Salton. *Automatic Information Organization and Retrieval.* McGraw-Hill Book Company, New York, 1968.

[29] Gerard Salton and Maria Smith. On the application of syntactic methodologies in automatic text analysis. In *Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 137–150, 1989.

[30] Tengku Mohd Tengku Sembok. *Logical-Linguistic Model and Experiments in Document Retrieval.* PhD thesis, Department of Computing Science, University of Glasgow, August 1989.

[31] A. F. Smeaton and C. J. van Rijsbergen. Experiments on incorporating syntactic processing of user queries into a document retrieval strategy. In *Eleventh International Conference on Research & Development in Information Retrieval*, pages 31–51, 1988.

[32] K. Sparck Jones and E. O. Barber. What makes an automatic keyword classification effective? *Journal of the American Society for Information Science*, pages 166–175, May-June 1971.

[33] K. Sparck Jones and R. G. Bates. Research on automatic indexing 1974 - 1976 (2 volumes). Technical report, Computer Laboratory. University of Cambridge, 1977.

[34] K. Sparck Jones and J. I. Tait. Automatic search term variant generation. *Journal of Documentation*, 40(1):50–66, March 1984.

[35] Karen Sparck Jones. *Automatic Keyword Classification for Information Retrieval.* Archon Books, 1971.

[36] Karen Sparck Jones. Collection properties influencing automatic term classification performance. *Information Storage and Retrieval*, 9:499–513, 1973.

[37] G. Thurmair. A common architecture for different text processing techniques in an information retrieval environment. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–143, 1986.

[38] Howard Turtle and W. Bruce Croft. Inference networks for document retrieval. In *Thirteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1990.

[39] Peter Willett. A fast procedure for the calculation of similarity coefficients in automatic classification. *Information Processing and Management*, 17:53–60, 1981.

[40] Clement T. Yu and Vijay V. Raghavan. Single-pass method for determining the semantic relationships between terms. *Journal of the American Society for Information Science*, pages 345–354, November 1977.