Automatic Phrase Indexing for Document Retrieval:

An Examination of Syntactic and Non-Syntactic Methods

Joel L. Fagan

Department of Modern Languages and Linguistics and
Department of Computer Science
Cornell University
Ithaca, New York 14853

Abstract

An automatic phrase indexing method based on the term discrimination model is described, and the results of retrieval experiments on five document collections are presented. Problems related to this non-syntactic phrase construction method are discussed, and some possible solutions are proposed that make use of information about the syntactic structure of document and query texts.

1. Introduction

In most fully automatic document retrieval systems, where both content analysis and retrieval are done without human intervention, documents and natural language queries are represented by an unstructured collection of simple descriptors (single words or word stems). Simple descriptors of this kind are not necessarily ideal content indicators. This is due, at least in part, to the fact that words vary widely in specificity. Some words are highly specific and therefore identify a narrow range of concepts, whereas other words are very general and may be associated with a broad range of concepts. For purposes of document retrieval, neither very specific nor very general descriptors are ideal, because they retrieve either too few or too many documents.

In order to improve the quality of an indexing vocabulary, it is often suggested that phrases be used as descriptors in place of (or in addition to) excessively general terms (Salton, Yang, and Yu 1975, Salton 1986, Dillon and Gray 1983).[1, 2] A number of methods have been proposed for automatically identifying phrases in the text of queries and documents. These phrase identification strategies range in complexity from simple procedures based on word frequencies and cooccurrence characteristics to rather sophisticated methods employing automatic syntactic analysis. Only a few of these methods, however, have been examined experimentally to determine their influence on retrieval effectiveness. Of those that have been tested experimentally, the method based on the term discrimination model has yielded increases in retrieval effectiveness that are among the best reported to date (Salton, Yang, and Yu 1975).

[1] The use of thesaurus classes is also advocated, but the current discussion is restricted to phrases.

[2] Dillon and Gray (1983) discuss this as a problem of ambiguity, but the problems addressed are essentially the same.

This study has three objectives: (1) to further investigate the level of effectiveness that can be achieved by a simple, non-syntactic phrase indexing strategy based on the term discrimination model, (2) to discuss some problems faced by this non-syntactic phrase indexing method, and (3) to suggest some solutions to these problems that make use of information about the syntactic structure of document and query texts.

2. Non-syntactic Phrase Indexing

2.1. Phrase Construction Method

The phrase indexing procedure used in the current study is based on the one proposed by Salton, Yang, and Yu (1975), but it has been generalized to test certain extensions to their original method. Because this method has been applied to five comparatively large document collections, and a large number of variations on the phrase construction process have been tested, the results provide an indication of the level of retrieval effectiveness attainable with this method that is more realistic than the results reported by Salton, Yang, and Yu (1975). The phrase construction process depends on the values specified for five parameters defined as follows:[3]

(1) **Length.** The maximum number of elements in a phrase. In these experiments, each phrase contains two elements.

(2) **Domain.** Elements of a phrase must occur together in a specified domain of cooccurrence. The domain may be a document (or query) or a sentence.

(3) **Proximity.** Elements of a phrase must occur within a specified proximity of one another within the specified domain of cooccurrence.

(4) **DFh.** A document frequency threshold for phrase elements is specified. The document frequency of term t, df_t, is defined as the number of documents in which term t occurs at least once. At least one element of each phrase must meet or exceed this threshold. This high document frequency element is called the *phrase head.*

(5) **DFp.** A document frequency threshold is specified. This threshold specifies that the document frequency of the phrase (not its elements) must meet or exceed a specified minimum value (DFp_{min}), or be less than a specified maximum value (DFp_{max}).

[3] As currently implemented, the phrase indexing procedure allows for substantial flexibility in specifying how phrases are to be constructed. Only those criteria for phrase construction that are directly relevant to the current discussion are mentioned here, however. Further details regarding the phrase construction method and analysis of retrieval results can be found in Fagan (to appear). The phrase indexing programs are written in C, and have been designed to interface easily with the SMART package (Buckley 1985).

The details of the phrase indexing procedure will be clarified by describing its application to the title of a sample document from the CISI collection. Using sentence as the domain of cooccurrence, a proximity of 1, DFh = 55, and $DFp_{min} = 1$ requires that: (1) the elements of a phrase cooccur in the same sentence, (2) the elements be adjacent in the document or query text after stopwords are removed, (3) each phrase contain at least one element having a document frequency of at least 55, and (4) the phrase occur in at least one document.

The first step of the indexing procedure is to identify individual word tokens in the text (see Figure 1), remove stopwords, and perform a stemming operation. At the same time, paragraph and sentence boundaries are recognized. The result of this step is illustrated in Figure 2. This information is used as the input to the phrase construction procedure. The column labeled "Phrase Head" in Figure 2 indicates the status of each token with regard to its acceptability as a phrase head, as determined by its document frequency. Phrase construction proceeds simply by combining phrase heads with adjacent tokens. For example, in Figure 2, the token *docu* is acceptable as a phrase head, so it is combined with adjacent tokens *associ* and *retrief* to form phrases *docu associ* and *docu retrief*. Similarly, the tokens *retrief* and *system* are both acceptable as phrase heads, and therefore combine to form the phrase *retrief system*. The order of phrase elements is regularized so that a pair of phrases cannot differ by order alone. Also, a phrase descriptor may not be constructed from two identical elements, so *word word* is not assigned as a phrase descriptor, even though the document frequency and proximity requirements for these tokens are met.

Figure 3 illustrates the final vector form of document 71. This vector consists of two subvectors: the single term subvector containing descriptors of type 0, and the phrase subvector containing descriptors of type 1. The weighting and similarity functions are described in the appendix.

.I 71
.T
Word-Word Associations in Document Retrieval Systems

FIGURE 1.
Original title of CISI document 71 (Lesk 1969).

Token	Descr Type	Doc Nbr	Para Nbr	Sen Nbr	Token Nbr	Doc Freq	Phrase Head
word	0	71	1	1	1	99	YES
word	0	71	1	1	2	99	YES
associ	0	71	1	1	3	23	no
docu	0	71	1	1	5	247	YES
retrief	0	71	1	1	6	296	YES
system	0	71	1	1	7	535	YES

FIGURE 2.
Input to phrase construction procedure
for the title of CISI document 71.

Document Number	Descriptor Number	Weight	Descriptor Type	Descriptor
71	26546	0.5706	0	associ
71	26850	0.2194	0	retrief
71	34344	0.7399	0	word
71	34406	0.2443	0	docu
71	39899	0.1380	0	system
71	10365	0.1787	1	retrief system
71	17459	0.2318	1	docu retrief
71	21114	0.6553	1	word associ
71	24244	0.4075	1	docu associ

FIGURE 3.
Final form of vector for the title of CISI document 71.

2.2. Retrieval Experiments

The objective of phrase indexing is to identify groups of words that will enhance retrieval effectiveness when assigned as phrase descriptors to representations of documents and queries. The non-syntactic approach to phrase indexing described above attempts to do this by taking into consideration five simple characteristics of words in text: (1) phrase length (2) domain of cooccurrence of phrase elements, (3) proximity of phrase elements, (4) document frequency of phrase elements, and (5) document frequency of phrases. These characteristics are incorporated into the phrase indexing procedure by specifying values for the parameters defined above. At the present time, there is no well-motivated basis for selecting values that can be expected to yield good retrieval results for a particular document collection. Thus in order to establish the level of retrieval effectiveness that can be attained with this method of phrase indexing, optimal values must be determined empirically for each experimental document collection. A large number of experiments have been conducted in which the phrase indexing procedure was applied repeatedly, while systematically varying parameter values. This was done for five document collections: CACM, INSPEC, CRAN, MED, and CISI. Basic characteristics of these collections appear in Table 1. For each set of values used, a retrieval experiment was done to compare the effectiveness of simple single term indexing to that of phrase indexing. In this way, a set of parameter values was established for each collection that yields optimal retrieval results for this phrase indexing method.[4]

Optimal parameter values for each collection appear in Table 2, together with retrieval evaluation results expressed as percent change in average precision in comparison to simple single term indexing. Tables 3 and 4 contain complete recall-precision results. These tables show that the responses of the test collections to the phrase indexing procedure were quite variable, both with respect to the level of retrieval effectiveness achieved, and optimal parameter values. Table 5 presents some additional average precision figures to illustrate how the different collections behave when identical parameter values are applied to all of them. These figures represent the results of only a small sample of the complete series of retrieval experiments. A variety of values for each phrase construction parameter have been tested. Experiments were done using both sentence and document as the domain of cooccurrence. Proximity values of 1-30, plus unlimited proximity, were tested. A continuum of values for the document frequency of phrase heads and the document frequency of phrases was tested until a clear indication of the effects that these parameters have on retrieval effectiveness could be perceived.

Characteristics	Collections				
	CACM	INSPEC	CRAN	MED	CISI
Number of Documents	3204	12684	1398	1033	1460
Number of Stem Types	4522	14255	3763	6927	5019
Mean Stems per Document	20.22	30.01	53.13	51.60	45.20
Number of Queries	52	77	225	30	76
Number of Stem Types	324	576	585	241	657
Mean Stems per Query	10.67	15.81	9.17	10.10	22.59

TABLE 1.
Statistics for document and query collections indexed with single terms, after stemming and stopword removal.

[4] It should be emphasized that the results of these experiments indicate an upper bound on retrieval performance for this indexing method, since the parameter values were selected to yield optimal results for these experimental document and query collections.

The figures in Table 5 show that the domain of cooccurrence has only a small influence on retrieval effectiveness. With unlimited proximity, for example, CACM has a 20.1% increase in average precision with a domain of document, and a 16.2% increase with a domain of sentence. For CISI the difference is somewhat larger, but still not great. With more restricted proximity values, varying the domain of cooccurrence has even smaller effects, and this holds true for all collections.

Proximity has a substantial influence on retrieval effectiveness for some collections, and an insignificant influence for others. Using document as the domain of cooccurrence, CACM shows an increase in average precision of 20.1% with unlimited proximity, and only a 7.6% increase with a proximity of 1. This is a difference of 12.5%. For MED, however, the same values yield a difference of only 1.4% (2.3% vs. 0.9%).

The general patterns revealed by testing various values for domain of cooccurrence and proximity are as follows. Three collections, CACM, INSPEC, and CRAN perform better when the relative location of phrase elements is completely unrestricted (domain: document, proximity: unlimited). CISI, however, performs best with maximally restrictive requirements for the relative location of phrase elements. The behavior of MED is perhaps most similar to that of CACM, INSPEC, and CRAN, since unlimited proximity is preferred. MED differs from these collections, however, in that a domain of sentence is preferred. For all collections, as the proximity of phrase elements increases, for the most part, gradual changes in average precision result. In particular, from a proximity of 5 upward, changes in average precision tend to be small. Since an increase in proximity causes more phrases to be assigned as descriptors, this suggests that a fairly balanced mix of good and bad phrases are added.

Experimentation with a wide range of values for the document frequency threshold for phrase heads (DFh) has revealed that this parameter has only a slight influence on retrieval effectiveness. The largest change in average precision due to this parameter was obtained for the CISI collection, with DFh = 50 (other values: domain: document, proximity: 1, $DFp_{min} = 1$). With DFh = 1, phrase indexing yielded a change in average precision of -2.2% in comparison to single term indexing (see Table 5), whereas with DFh = 50, the change increased very slightly to -1.5%. Other collections showed either decreases in average precision, or even smaller positive changes with values of DFh greater than one.

The document frequency threshold for phrases was used to test the effect of excluding both low and high document frequency phrases from use as phrase descriptors. The motivation for excluding low document frequency phrases is that phrases that occur in a very small number of documents stand a good chance of simply being fortuitous cooccurrences of terms rather than meaningful phrases. Low document frequency phrases are excluded by specifying a value for DFp_{min}, and then assigning phrase p as a descriptor only if $df_p \geq DFp_{min}$. The largest increase in average precision was obtained for the MED collection. With $DFp_{min} = 1$ (other parameter values: domain: sentence, proximity: unlimited, DFh = 1), phrase indexing yielded a change in average precision of +3.3% over single term indexing. With $DFp_{min} = 3$, this increased very slightly to +4.0%. A substantial change in average precision due to exclusion of low document frequency phrases was obtained only for CACM. With $DFp_{min} = 1$ (other parameter values: domain: document, proximity: unlimited, DFh = 1), phrase indexing yielded a change in average precision of +20.1% over single term indexing. With $DFp_{min} = 2$, this dropped to +14.0%, for a decrease of 6.1%. In general, excluding phrases with document frequencies of 3 or less has a very slight positive effect for some collections, and a negative effect for other collections. For all collections tested, as DFp_{min} is increased above 3, average precision declines steadily.

The motivation for excluding high document frequency phrases has to do with the fact that high document frequency phrases tend to contain high document frequency single terms, and that high document frequency single terms tend to have a negative effect on precision. Matches on high document frequency phrases (in addition to the corresponding single terms) thus tend to emphasize the negative effect of the high document frequency single terms. High document frequency phrases are excluded by specifying a value for DFp_{max} and then assigning phrase p as a descriptor only if $df_p < DFp_{max}$. A comparison of the best average precision values for each collection in Table 5 with those in Table 2 reveals that by excluding high document frequency phrases, small increases in average precision can be achieved. Using values for DFp_{max} of 90, 150, 90, and 30 for CACM, INSPEC, CRAN, and CISI (respectively), results in increases in average precision of 0.7% to 4.0% over the best average precision values in Table 5.

Some conclusions can be drawn regarding the general applicability of this phrase indexing method:

(1) Though this approach to phrase indexing can have a substantial positive effect on retrieval effectiveness, the method does not consistently result in substantial improvements in effectiveness for all collections. This is indicated clearly by the range of increases in average precision for the five collections: 2.2% to 22.7%. According to the criteria suggested by Sparck Jones (1974: 397), a material improvement is achieved only by CACM and INSPEC. CRAN and MED show lower levels of statistically significant improvement, and CISI shows only a slight, statistically insignificant increase in average precision.[5]

Collection	Non-syntactic Phrase Indexing Parameters				Average Precision Change	Statistically Significant Change?	Material Change?
	Domain	Proximity	Phrase Head Document Frequency (DFh)	Phrase Document Frequency (DFp)			
CACM	document	unlimited	1	< 90 (0.03n)	+22.7%	yes $P < 0.01$	yes
INSPEC	document	unlimited	1	< 150 (0.01n)	+11.9%	yes $P < 0.01$	yes
CRAN	document	unlimited	1	< 90 (0.06n)	+8.9%	yes $P < 0.01$	no
MED	sentence	unlimited	3*	≥ 3	+4.0%	yes $P < 0.01$	no
CISI	sentence	1	1	< 30 (0.02n)	+2.2%	no $P > 0.05$	no

TABLE 2.

Best parameter values and summary of retrieval results. Average precision change is with respect to single term indexing; see Tables 3 and 4. The value n is collection size; see Table 1.

* This value is a by-product of the threshold for the document frequency of phrases, rather than an independently specified requirement.

Recall Level	Precision					
	CACM		INSPEC		CRAN	
	Single Term Indexing	Phrase Indexing	Single Term Indexing	Phrase Indexing	Single Term Indexing	Phrase Indexing
0.00	0.5709	0.7613	0.6634	0.7505	0.7758	0.8206
0.05	0.5545	0.7315	0.5852	0.6885	0.7758	0.8206
0.10	0.5086	0.6489	0.5261	0.6084	0.7526	0.8001
0.15	0.4822	0.5988	0.4628	0.5386	0.6900	0.7362
0.20	0.4343	0.5335	0.4181	0.4923	0.6187	0.6704
0.25	0.4073	0.5060	0.3810	0.4400	0.5521	0.6065
0.30	0.3672	0.4542	0.3412	0.3893	0.5184	0.5659
0.35	0.3184	0.3911	0.3008	0.3399	0.4448	0.4879
0.40	0.2972	0.3569	0.2781	0.3090	0.4282	0.4732
0.45	0.2573	0.3199	0.2462	0.2782	0.3822	0.4235
0.50	0.2398	0.2971	0.2283	0.2488	0.3714	0.4116
0.55	0.2064	0.2572	0.2045	0.2118	0.3094	0.3431
0.60	0.1912	0.2416	0.1777	0.1900	0.2952	0.3240
0.65	0.1726	0.2089	0.1581	0.1606	0.2734	0.2953
0.70	0.1462	0.1719	0.1360	0.1380	0.2301	0.2452
0.75	0.1323	0.1541	0.1150	0.1150	0.2107	0.2283
0.80	0.1086	0.1261	0.0936	0.0942	0.1839	0.2001
0.85	0.0860	0.0909	0.0643	0.0690	0.1553	0.1713
0.90	0.0711	0.0742	0.0484	0.0527	0.1313	0.1474
0.95	0.0619	0.0624	0.0293	0.0315	0.1179	0.1311
1.00	0.0610	0.0615	0.0179	0.0199	0.1175	0.1307
Avg Prec	0.2604	0.3195	0.2459	0.2750	0.3852	0.4194
% Change		22.7		11.9		8.9

TABLE 3.

Comparison of single term (stem) indexing and phrase indexing. Phrase indexing parameters as given in Table 2.

Recall Level	Precision			
	MED		CISI	
	Single Term Indexing	Phrase Indexing	Single Term Indexing	Phrase Indexing
0.00	0.9254	0.9535	0.6656	0.6838
0.05	0.8853	0.9082	0.5623	0.5769
0.10	0.8036	0.8512	0.4919	0.4947
0.15	0.7620	0.8061	0.4366	0.4385
0.20	0.7258	0.7843	0.4032	0.4026
0.25	0.6964	0.7676	0.3668	0.3720
0.30	0.6742	0.7222	0.3118	0.3285
0.35	0.6563	0.6720	0.2872	0.2947
0.40	0.6317	0.6430	0.2624	0.2712
0.45	0.5956	0.6052	0.2463	0.2528
0.50	0.5447	0.5570	0.2320	0.2330
0.55	0.4985	0.5107	0.2057	0.2080
0.60	0.4728	0.4818	0.1901	0.1982
0.65	0.4441	0.4444	0.1787	0.1858
0.70	0.4082	0.4175	0.1504	0.1556
0.75	0.3825	0.3871	0.1312	0.1333
0.80	0.3501	0.3536	0.1119	0.1131
0.85	0.2900	0.2947	0.0843	0.0919
0.90	0.2057	0.2127	0.0739	0.0811
0.95	0.1234	0.1286	0.0601	0.0661
1.00	0.0888	0.0970	0.0521	0.0582
Avg Prec	0.5378	0.5595	0.2450	0.2503
% Change		4.0		2.2

TABLE 4.

Comparison of single term (stem) indexing and phrase indexing. Phrase indexing parameters as given in Table 2.

(2) A single phrase selection strategy is not effective for all collections. CACM, INSPEC, and CRAN perform best when very unrestrictive phrase selection criteria are employed, that is, with the broadest domain of cooccurrence, and unlimited distance between phrase elements. MED can be grouped with CACM, INSPEC, and CRAN, since it performs best with the least restrictive proximity requirement. In contrast, CISI performs best with maximally restrictive phrase selection criteria, where phrase elements must cooccur adjacently in the same sentence. For all collections, further restrictions on the document frequency of phrases and phrase elements (heads and components) have only a slight effect on retrieval performance.

[5] The Wilcoxon signed rank test for paired observations was used to test the significance of the differences between average precision values.

Proximity	CACM	INSPEC	CRAN	MED	CISI
Domain: Document					
unlimited	**0.3128 +20.1%**	**0.2652 +7.9%**	**0.4169 +8.2%**	0.5501 + 2.3%	0.2167 -11.5%
1	0.2803 +7.6%	0.2546 +3.5%	0.3989 +3.6%	0.5429 +0.9%	0.2396 -2.2%
Domain: Sentence					
unlimited	0.3025 +16.2%	0.2534 +3.0%	0.4105 +6.6%	**0.5555 +3.3%**	0.2326 -5.0%
1	0.2808 +7.9%	0.2545 +3.5%	0.3991 +3.6%	0.5435 +1.1%	**0.2406 -1.8%**

TABLE 5.

Average precision change using identical phrase indexing parameter values for all collections. Average change is with respect to single term indexing, see Tables 3 and 4. The best value for each collection in this table is given in boldface. Other parameter values are: length 2, DFh: 1, DFp_{min}: 1.

(3) Because the domain of cooccurrence and the document frequency of phrases and phrase heads have such a small influence on retrieval effectiveness, a level of performance that approximates the optimal level shown in Table 2 can be achieved by simply disregarding the document frequency and domain parameters. Thus a very simple approach to phrase indexing that makes use of only the length and proximity parameters can be expected to perform reasonably for all collections. For practical purposes, this eliminates the need to determine acceptable document frequency thresholds experimentally for each collection.

(4) The information about term specificity and relationships among words in text that is provided by document frequency and the relative location of words in text does not provide an adequate basis for a phrase indexing procedure that will consistently yield substantial, statistically significant improvements in retrieval effectiveness. This suggests that more detailed information about text structure and relationships among words is required. Sections 3 and 4 discuss this idea further.

2.3. Comparison with Other Phrase Indexing Experiments

This section compares the performance of the phrase indexing procedure described above with four previous experiments in phrase indexing: Salton, Yang, and Yu's (1975) work based on the discrimination value model, Dillon and Gray's (1983) procedure which is based on dictionary look-up of syntactic patterns, and Croft's (1986) and Smeaton's (1986) studies of term dependencies derived manually from syntactically correct natural language query phrases.[6]

The results of Salton, Yang, and Yu's (1975) experiments are presented in summary form in Table 6. This table contains two sets of comparisons for three small document collections. The row labeled "tf" compares the average precision attained with simple single term indexing with term frequency (tf) weights to the results of phrase indexing. This is the data provided by Salton, Yang, and Yu (1975), and it shows that phrase indexing yields an increase in average precision of between 17% and 39% over single term indexing with tf weights. The row labeled "tf×idf" compares the same phrase indexing results to results of single term indexing with weights calculated as a product of term frequency and inverse document frequency (tf×idf). These figures are based on results reported by Salton and Yang (1973). In comparing phrase indexing with single term indexing and this better weighting method, their phrase indexing still shows an increase in average precision, but the magnitude of the increase is much less, ranging between 6% and 20% rather than 17% and 39%. The tf×idf figures provide a better point of comparison with the current study, since single term indexing with tf×idf weights is used as the basic point of reference for evaluating retrieval results.

A comparison of these results with the figures in Table 2 reveals that Salton, Yang, and Yu's results are comparable to, or better than, the results obtained in this study. In the present study, the best average precision increase was 22.7% for CACM. This is comparable to Salton, Yang, and Yu's result for the MEDLARS collection. The average precision increases obtained for INSPEC and CRAN are close to the 11% increase obtained by Salton, Yang, and Yu for the CRANFIELD collection and their 6% increase for the TIME collection. The results obtained in this study for CISI and MED, however, are lower than any of the results obtained by Salton, Yang, and Yu.

Dillon and Gray (1983) compared a single term indexing procedure using stemming and stopword removal to a syntactic phrase indexing procedure based on matching sequences of syntactic categories assigned to text words against a dictionary of patterns of syntactic categories. Inverse document frequency (idf) weights and the cosine similarity function were used for retrieval with both indexing methods. The two indexing procedures were applied to a collection of 250 library science master's papers and 22 natural language queries.[7] The phrase indexing procedure proved to be slightly better than the single term procedure as indicated by increases in precision at most recall levels below 80%. At 40-60% recall, the increase in precision ranged between 3% and 7%.

These results are not directly comparable to the non-syntactic phrase indexing procedure of the present study, since different collections were used. However, for CACM, INSPEC, and CRAN the non-syntactic procedure appears to be as good as, or better than, Dillon's syntax-based procedure. It may be that Dillon's procedure would yield better results if used as a supplement to a simple single term indexing procedure rather than as an alternative, since preliminary experiments with non-syntactic phrase indexing have shown that excluding single term descriptors is detrimental to retrieval performance.

Croft (1986) and Smeaton (1986) have experimented with methods of incorporating information about natural language query phrases into the retrieval process. In both cases, term dependencies are derived manually from the natural language text of queries, so the dependencies correspond closely to syntactically correct natural language phrases. Dependencies are then incorporated into the retrieval process by increasing the retrieval rank of a particular document if that document contains the elements of a set of dependent query terms. For Croft, the dependent terms may occur anywhere in the document. Smeaton's cooccurrence requirements are stricter, in that dependent terms must cooccur within a sentence, clause, or phrase in the document text.

Both Croft and Smeaton performed retrieval experiments on the CACM collection, comparing their term dependency methods to single term indexing with idf weights. Croft's single term retrieval yielded an average precision of 0.2110. After taking into consideration dependent terms, the average precision rose to 0.2270, for an increase of 7.6%.[8] Using a collection of 25 queries, Smeaton's single term indexing resulted in an average precision of 0.2223. His best

[6] In making comparisons of the kind presented here, it should be remembered that in general it is difficult to draw firm conclusions about the relative effectiveness of indexing and retrieval methods that have been tested in different laboratories or at different times, since experimental conditions may differ significantly. Document and query collections differ, and the details of even widely accepted indexing, retrieval, and evaluation procedures may differ. Nevertheless, it is still instructive to compare experimental results in order to get a general idea of the relative effectiveness achieved by different indexing and retrieval methods, provided that the limitations of such comparisons are kept in mind.

[7] A third, thesaurus-based procedure was also tested, but its performance was much worse than the others.

[8] These figures are based on data in Croft (1986), Table 2, p. 75. Average precision was calculated at recall levels 0.10-0.90.

Single Term Weighting Method	Average Precision					
	CRANFIELD 424 documents		MEDLARS 450 documents		TIME 425 documents	
	Single Term Indexing	Phrase Indexing	Single Term Indexing	Phrase Indexing	Single Term Indexing	Phrase Indexing
		0.4287		0.5468		0.6783
tf	0.3207	+32%	0.4158	+39%	0.5794	+17%
tf×idf	0.3788	+11%	0.4722	+20%	0.6440	+6%

TABLE 6.
Comparison of average precision for single term (stem) indexing using two weighting methods (tf and tf×idf) and Salton, Yang and Yu's phrase indexing method. Percentages indicate changes in average precision attained by phrase indexing. Figures for the tf weights are from Salton, Yang, and Yu (1975); those for the tf×idf weights are based on Salton and Yang (1973).

term dependency method then increased this to 0.2754, for an improvement of 23.9%.[9]

In comparing these results to the non-syntactic method of the present study, it is important to note that Smeaton and Croft compared their term dependency results to single term indexing with idf weights, rather than tf×idf weights. Table 7 compares retrieval results for the CACM collection for single term indexing using two different weighting methods, idf and tf×idf. Two different query collections were tested: (1) the 25 queries used by Smeaton,[10] and (2) the complete collection of 52 queries. Smeaton's 25 queries are a subset of the larger collection. This table shows that single term indexing with tf×idf weights performs better than single term indexing with idf weights for both query collections. Using single term indexing with tf×idf weights as a point of comparison thus provides a more demanding basis for evaluating the performance of a phrase indexing method.

As can be seen from Table 8, the non-syntactic phrase indexing procedure yields an increase in average precision of 38.3% over single term indexing with tf×idf weights when applied to Smeaton's set of 25 queries. For the set of 52 queries, the non-syntactic procedure yields an increase of 22.7% over single term indexing with tf×idf weights (see also Tables 2 and 3). A clearer indication of the potential benefits of Croft's and Smeaton's approaches could be obtained by applying their methods to additional document and query collections.

In summary, it can be concluded that: (1) when applied to the CACM collection, the non-syntactic procedure yields an increase in average precision that is at least competitive with Smeaton's syntax-based procedure, and better than Croft's method; (2) at its best, the non-syntactic method exceeds the increases in precision achieved by Dillon's syntax-based method, but on other collections, the performance may be approximately equal, or slightly inferior; and (3) the results obtained by Salton, Yang, and Yu are comparable to some of the results of the present study (CACM, INSPEC, CRAN), and better than others (MED, CISI). The results of the present study, however, provide a more realistic indication of the level of retrieval effectiveness that can be achieved using a non-syntactic phrase indexing procedure, since the experiments were performed using more, and significantly larger, document and query collections than were used by Salton, Yang, and Yu.

[9] This calculation is based on Smeaton's data for single term indexing and his phrase indexing method that yields the best average precision figures. The data is given in Smeaton (1986), Table 8, columns labeled "IDF Uncorrected" and "Corrected 5,10,5,10." Averages were calculated at recall levels 0.10-0.90.

[10] I am grateful to Alan Smeaton for providing me with information about the query collection used for his experiments.

Query Collection	Single Term Indexing idf weights	Single Term Indexing tf×idf weights
Smeaton's 25	0.2079	0.2230 +7.3%
Standard 52	0.2153	0.2604 +21.0%

TABLE 7.
Comparison of average precision for the CACM collection using single term indexing with two weighting methods and two query collections. (Precision averages calculated at recall levels 0.10-0.90.)

Query Collection	Single Term Indexing tf×idf weights	Non-syntactic Phrase Indexing
Smeaton's 25	0.2230	0.3083 +38.3%
Standard 52	0.2604	0.3195 +22.7%

TABLE 8.
Comparison of average precision using single term indexing and phrase indexing for the CACM document collection and two query collections. The weighting method for phrase indexing is described in the appendix. Phrase indexing parameter values are those given in Table 2. (Precision averages calculated at recall levels 0.10-0.90.)

3. Problems with Non-syntactic Phrase Indexing

In examining large samples of phrases generated by the non-syntactic procedure and the effect they have on retrieval performance, a number of problems with this procedure have become apparent. This section discusses a few of these problems. It should be emphasized that examples like the ones given below are not isolated occurrences. Similar cases occur abundantly in the document and query collections.[11] Section 3.1 examines several examples of phrases identified by the procedure that ideally should not be used as phrase descriptors. Section 3.2 discusses cases in which natural language phrases that are good indicators of document content can-

[11] All examples are taken from experimental document and query collections. The source is given in parentheses after each example. That is, (CISI q12) and (CISI d1340) refer to query 12 and document 1340 in the CISI collection.

not be identified as phrase descriptors due to limitations of the non-syntactic phrase indexing procedure. For illustrative purposes, this discussion assumes a relatively restrictive non-syntactic phrase construction strategy using parameter settings such as those given for the CISI collection in Table 2.

3.1. Construction of Inappropriate Phrase Descriptors

A phrase descriptor may be thought of as inappropriate for two general reasons. First, the descriptor may simply not be an accurate indicator of document or query content. Second, the meaning of the source text of a phrase descriptor in a query may differ significantly from the meaning of the source text of a phrase descriptor in a document.

Phrase indexing consists of two processes: (1) identification of phrases in text, and (2) normalizing the form of phrases that differ in structure, but that may be related in meaning. Normalization is beneficial, since it makes it possible to represent a pair of phrases like *information retrieval* and *retrieval of information* by the single phrase descriptor *inform retrief*. Similarly, the phrases *book review* and *reviews of books* can both be represented by the phrase descriptor *book review*. In non-syntactic phrase indexing, normalization is accomplished by three devices: (1) stemming, (2) regularizing the order of phrase elements, and (3) ignoring stopwords that intervene between content words. All of these devices must be used in order to accomplish the normalization just illustrated. While normalization has significant benefits, many of the inappropriate phrase descriptors generated by the non-syntactic phrase indexing process are the result of excessive normalization.

Seven queries for the CACM collection contain the text phrase *operating system*, which yields the phrase descriptor *oper system*. In all of these queries, the source of this descriptor is syntactically correct, and the descriptor is a good indicator of document content. A number of documents contain this descriptor, but many of them are related only peripherally, if at all, to the topic of operating systems. The phrase descriptor *oper system* does not correspond to a single phrase in document and query texts, or even to a set of phrases closely related in meaning. For example:

(1) a fully automatic document retrieval *system operating* on the IBM 7094 is described (CACM d1236)
(2) to illustrate *systems operations* and evaluation procedures (CACM d1236)
(3) extensive data on the *system's operation* (CACM d1533)
(4) to achieve a *system operational* within six months (CACM d2380)
(5) time between project inception and *system operational* date (CACM d1034)
(6) critical to the *system's operating* efficiency (CACM d1226)
(7) examples of overall *system operation* (CACM d3087)
(8) the *system, operated* entirely from a digital display unit, interacts directly with the user (CACM d1695)
(9) the *system* is *operational* and available on the arpa sdc time shared computing system (CACM d1170)
(10) the *system* has been in *operation* (CACM d1665)
(11) the COBOL language was used specifically to enable the *system* to *operate* on three IBM computers (CACM d1168)
(12) the logic required in procedures, *operations, systems,* and circuits (CACM d320)
(13) examples of the *operation* of *system* components (CACM d3087)
(14) an *operational system* utilizing this concept (CACM d2919)
(15) the duplex *operation* gives the *system* greater reliability (CACM d252)

In all of these examples, the inappropriate document phrase descriptors are the result of a pair of words that happen to occur in close proximity in the text, but that nevertheless are not related syntactically. That is, they do not enter into a relationship of modification with one another. The result is a phrase descriptor that does not accurately reflect the content of the documents, and that matches with a query descriptor whose source text differs significantly in meaning from the sources of all of the document descriptors. The net effect of this phrase descriptor on retrieval performance for CACM is a reduction in average precision.

3.2. Failure to Identify Good Phrase Descriptors

This section illustrates a common situation in which the non-syntactic phrase indexing process fails to identify phrase descriptors that are good content indicators, and that should have a positive influence on retrieval performance. Whereas simple frequency and proximity criteria often fail to identify useful phrase descriptors, relatively simple syntactic criteria can be used to successfully recognize many appropriate phrases.

Noun phrases involving conjunctions are an important source of good phrase descriptors that cannot adequately be identified on a non-syntactic basis. Consider, for example, the text phrase

> parallel and sequential algorithms (CACM q63).

Non-syntactic indexing yields from this the correct phrase *sequential algorithm*, and the meaningless *parallel sequential*. Syntactic analysis provides the information that both *parallel* and *sequential* can be understood as modifiers of *algorithms*, thus making it possible to generate two correct phrases, *parallel algorithm* and *sequential algorithm*, and to avoid the inappropriate phrase identified by the non-syntactic procedure.

The same strategy can be applied to more complex constructions. For example, from the text phrase

> the structure, analysis, organization, storage, searching, and retrieval of information (CISI d175),

the non-syntactic phrase construction process identifies the phrases

*structure analysis	*storage searching
*analysis organization	*searching retrieval
*organization storage	information retrieval

Five of the six phrases generated are not good indicators of document content; these are indicated by asterisks. In contrast, simple syntactic information makes it possible to generate

information structure	information storage
information analysis	information searching
information organization	information retrieval,

and to avoid constructing all of the inappropriate phrases listed above.

The abundance of constructions of this kind in titles and abstracts is a strong indication that a large number of good phrase descriptors could be identified using syntactic criteria that could not be identified on the basis of frequency and cooccurrence criteria alone. The following section proposes a method for incorporating information about syntactic relationships among words into the phrase construction process.

4. Syntactic Phrase Indexing

The objective of syntax-based phrase construction is to use information about the syntactic structure of document and query texts to identify relationships among words that will make it possible to construct useful phrases that could not be correctly identified without syntactic information, and to avoid constructing inappropriate phrases that would be generated with a non-syntactic procedure.

A prototype phrase construction system has been implemented using PLNLP,[12] (Heidorn 1972, 1975) and a broad-coverage computational grammar of English which is also written in PLNLP (Jensen 1986). Based on the structural information produced by this syntactic analysis system, the phrase construction procedure decomposes complex constructions into simpler forms, while preserving much of the information about syntactic relationships among words. In addition, the procedure normalizes the form of constructions that differ syntactically, but that are closely related semantically. This is done in such a way that the resulting phrases can be incorporated directly into the vector representation of documents and queries, thus maintaining compatibility with the existing retrieval environment.

[12] Programming Language for Natural Language Processing.

To illustrate the essential objectives and strategy of this approach, consider the following set of natural language phrases:

text analysis	automatic text analysis
analysis of text	automatic analysis of text
analysis of scientific text	automatic analysis of scientific text
analysis of literary text	automatic analysis of literary text

All of these phrases are related semantically, having to do with text analysis. They have several words in common, but differ in syntactic structure. If these phrases occurred in different documents in a collection, and a user submitted a general query containing a phrase like *automatic text analysis*, the retrieval system should have the capability of recognizing some similarity between the query and all of these phrases.

Trees (T1) and (T2) are parse trees produced by the syntactic analyzer. Given this syntactic information, phrases are constructed by simply combining the head of a constituent with the head of each constituent that modifies it. In the parse trees, an asterisk indicates the head of each constituent. In (T1), the noun *analysis* is the head of the complete noun phrase, and it has two premodifiers, an adjective phrase and a noun phrase. These modifying constructions have the adjective *automatic* and the noun *text* as heads. The modifier *automatic* is combined with *analysis* to yield the phrase *automatic analysis*. Similarly, *text* is combined with *analysis* to yield the phrase *text analysis*.

(T1) automatic text analysis

NP	AJP	ADJ*	automatic
	NP	NOUN*	text
	NOUN*	analysis	

Yields: automatic analysis, text analysis

In (T2), *analysis* is again the head of the complete noun phrase, and *automatic* is the head of the premodifying adjective phrase. In this case, however, *text* is the head of a prepositional phrase postmodifier, rather than a noun phrase premodifier. Nevertheless, the same strategy illustrated for (T1) applies here to construct the same two phrases, *automatic analysis* and *text analysis*. Within the prepositional phrase, *scientific* is the head of an adjective phrase modifying *text*, so this head and modifier combine to form *scientific text*. Note also that the incorrect phrase **scientific analysis* is not produced, since these words are not related as head and modifier. This incorrect phrase would be generated by the non-syntactic procedure.

(T2) automatic analysis of scientific text

NP	AJP	ADJ*	automatic	
	NOUN*	analysis		
	PP	PREP	of	
		AJP	ADJ*	scientific
		NOUN*	text	

Yields: automatic analysis, text analysis, scientific text
Avoids: *scientific analysis

By applying this procedure of decomposition and normalization to the eight natural language phrases listed above, they can all be represented by some combination of the following four simpler phrases:

Modifier	Head
text	analysis
scientific	text
literary	text
automatic	analysis

After application of a stemming operation, these phrases would be included as elements of the phrase subvector.

Tree (T3) is a more complex noun phrase that further illustrates the benefits of using syntactic information for phrase construction.[13] This is a noun phrase that consists of two conjoined noun phrases. The second conjunct has a prepositional phrase postmodifier that has a conjunction of noun phrases as object. With noun phrases having this pattern of modification, a number of useful phrases can be generated by applying the same basic decomposition procedure. For purposes of phrase construction, the postmodifying prepositional phrase is treated as a modifier of both elements of the conjunction *preparation and evaluation*. Each of the conjuncts of *abstracts and extracts* is therefore combined with each of the conjuncts of *preparation and evaluation* to produce four phrases: *abstracts preparation, extracts preparation, abstracts evaluation,* and *extracts evaluation.*

Proceeding into the lower levels of the parse tree, the adjectival premodifier *computer-prepared* is combined with both elements of the conjunction *abstracts and extracts* to produce the phrases *computer-prepared abstracts,* and *computer-prepared extracts.* Before constructing the final vector representation of the document, each word is reduced to its base form, in order to normalize inflectional and derivational variants.

Again, by using syntactic information it is possible to avoid constructing inappropriate phrases like **computer-prepared evaluation*, **preparation evaluation*, and **abstracts extracts*, all of which would be generated by the non-syntactic procedure.

So far, the phrase construction rules have been implemented only for generating phrase descriptors from noun phrases. However, work is in progress that will extract phrases from verbal constructions, and also normalize the form of certain nominal and verbal expressions. For example, in the sentence

This is a system for **automatic analysis of text.**

the noun *analysis* has an adjectival premodifier *automatic* and a prepositional phrase post-modifier *of text.* But in the sentence

This is a system that **analyzes text automatically.**

the same idea is expressed in a relative clause with *text* as the objective of the verb *analyze*, which is modified by the adverb *automatically.* Normalization of constructions of this kind will further enhance the phrase indexing procedure.

The phrase construction procedure as outlined here is clearly in need of refinements and extensions. One of the most obvious problems that must be dealt with is the structural ambiguity of complex noun phrases. Though it is not possible to resolve ambiguities of this kind given syntactic information alone, examination of sample document and query texts indicates that it should be possible to develop rules for phrase construction that will deal adequately with a large proportion of cases. After a more fully developed set of phrase generation rules are in place, retrieval experiments will be conducted to compare the level of effectiveness achieved by non-syntactic phrase indexing with that of the approach to syntactic phrase indexing outlined here.

5. Summary and Conclusion

(1) The automatic phrase construction procedure based on relative location of phrase elements and on term specificity (as indicated by the document frequency of phrases and phrase elements) does not consistently yield substantial improvements in retrieval effectiveness when applied to five experimental document collections. Thus it is not likely that phrase indexing of this kind will prove to be an important method of enhancing the performance of automatic document indexing and retrieval systems.

(2) Even though the non-syntactic method does not consistently perform well, when applied to the CACM collection, it yields improvements in retrieval effectiveness that are at least competitive with Croft's and Smeaton's methods that are based on syntactically correct natural language phrases from queries. This unexpected result may be due, at least in part, to the fact that those methods do not take into consideration syntactic relationships among words in documents. The examples discussed in section 3.1 show clearly that simple cooccurrence in a document of the elements of a syntactically correct query phrase does not guarantee the presence of a document phrase that is closely related in meaning to the query phrase. Many of the inappropriate phrase matches that occur with the non-

[13] Additional examples were presented in Fagan (1985).

(T3) the preparation and evaluation of computer-prepared abstracts and extracts (CISI d6).

NP	DET	ADJ*	the			
	NP	NOUN*	preparation			
	CONJ*	and				
	NP	NOUN*	evaluation			
		PP	PREP	of		
			NP	AJP	ADJ*	computer-prepared
				NOUN*	abstracts	
			CONJ*	and		
			NP	NOUN*	extracts	

Yields: abstracts preparation, extracts preparation, abstracts evaluation, extracts evaluation computer-prepared abstracts, computer-prepared extracts

Avoids: *computer-prepared evaluation, *preparation evaluation, *abstracts extracts

syntactic method are therefore likely to occur with Smeaton's and Croft's methods as well. Thus in order to take full advantage of the possible enhancements that syntactic information might provide, it appears that syntactic relationships among words in both queries and documents must be taken into consideration.

(3) The automatic syntactic phrase construction process outlined in section 4 is capable of overcoming many of the difficulties encountered in non-syntactic phrase construction, as described in section 3. Since this approach is integrated into a broad-coverage automatic natural language analysis system, it can be applied to both documents and queries, and therefore should provide for more reliable matches between query and document phrases.

(4) It has long been acknowledged that information about relationships among terms in text should be exploited as a source of improvement in automatic document analysis and retrieval systems. Research on this general topic began with the early work on statistical term associations by Stiles (1961), Doyle (1961, 1962), Giuliano and Jones (1963), Salton (1968), and Lesk (1969), as well as early work on syntax-based approaches (Baxendale 1958, 1961; Salton 1966; Earl 1970, 1972; Hillman and Kasarda 1969; Hillman 1973; Klingbiel 1973a, b). Investigation has continued with more recent work on probabilistic terms dependency models (van Rijsbergen 1977, Harper and van Rijsbergen 1978, Yu et al. 1983, Salton, Buckley and Yu 1983), and syntactic methods (Dillon and Gray 1983, Aladesulu 1985, Metzler et al. 1984, Smeaton 1986). Nevertheless, there is still no well-established consensus regarding how information about term relationships should be obtained and incorporated into document retrieval systems, or the extent to which this kind of information can be expected to yield consistently positive results. The experimental results of the present study and the analysis of problems related to non-syntactic phrase construction methods indicate clearly, however, that if information about term relationships is to be used in a way that will yield significant improvements, then it will be necessary to go beyond simple measures of term frequencies, cooccurrence characteristics, and proximity in analyzing text structure and identifying relationships among words. Thus careful experimental evaluation of a syntactic phrase indexing strategy like the one outlined in section 4 should provide valuable insights regarding the potential benefits of automatic phrase indexing for document retrieval.

Acknowledgments

I am indebted to Gerard Salton for the advice he has provided in planning and carrying out these experiments, and for his comments on a number of drafts of this paper. The help that Chris Buckley and Ellen Voorhees provided in implementing the non-syntactic phrase indexing software, and their advice on other aspects of this work are greatly appreciated. I am also grateful to George E. Heidorn and Karen Jensen of IBM Thomas J. Watson Research Center for providing the facilities and background required to develop the syntax-based phrase indexing system.

This work was supported in part by OCLC, Inc. and the National Science Foundation (grant IST 83-16166).

Appendix

Weighting and Similarity Functions

The weight assigned to a descriptor in a vector is indicative of the importance of the descriptor as an indicator of document or query content. In order to include information about the relative importance of a term in an individual document or query, the weighting function used in these experiments incorporates the frequency of each term in a given document or query. As an indication of the quality of a descriptor with respect to the document collection as a whole, the inverse document frequency ratio is included. A discussion of these two weighting factors can be found in Sparck Jones (1972) and Salton and Yang (1973). Finally, the cosine normalization is used in order to normalize for vector length.

The following expressions define the weighting function. Initially, the weight of term t in vector v is the frequency of t in the document or query represented by v. This is a simple term frequency weight, tf_{tv}. The term frequency weights are normalized by dividing by the maximum term frequency in the vector, max_tf_v:

$$norm\_tf_{tv} = \frac{tf_{tv}}{max\_tf_v} \tag{1}$$

The inverse document frequency ratio is incorporated with the definition given in (2),

$$tf\_idf_{tv} = norm\_tf_{tv} \cdot \ln\frac{n}{df_t} \tag{2}$$

where n is the number of documents in the collection, and df_t is the document frequency of t, that is, the number of documents in which term t occurs at least once.

The cosine normalization yields the final weight, w_{tv}, of term t in vector v:

$$w_{tv} = \frac{tf\_idf_{tv}}{\sqrt{\sum_{i=1}^{k} tf\_idf_{iv}^{\,2}}} \tag{3}$$

where k is the length of vector v.

The weights defined by expressions (1)-(3) are used for single term descriptors in collections that are indexed with single terms only, as well as for single term descriptors in collections indexed with both single terms and phrases. In collections indexed with both single terms and phrases, however, normalization is done over the single term subvector only, rather than over the entire vector. Thus the single term subvector for a document (or query) in a collection indexed with single terms and phrases is identical to the vector for the same document (or query) in a collection indexed with single terms only.

The weight of a phrase descriptor is a function of the weights of its elements. That is, if phrase p in vector v is composed of single terms a and b, also in vector v, then the weight, w_{pv}, of phrase p in vector v is:

$$w_{pv} = \frac{w_{av} + w_{bv}}{2} \tag{4}$$

This phrase weight has been chosen for two reasons. First, since the phrase weight is a function of the weights of the phrase elements, it incorporates information about the importance of the elements of the phrase into the phrase weight. Second, it assures that the magnitude of phrase weights does not differ greatly from the magnitude of single term weights.

A document or query indexed with both single terms and phrases consists of two subvectors, one containing single term descriptors, and one containing phrase descriptors.[14] In order to calculate the similarity between a query vector and a document vector, a partial similarity is calculated for each subvector, and the overall similarity is then calculated as a weighted sum of the two partial similarities.

Let q represent a query vector consisting of a single term subvector q_s and a phrase subvector q_p; similarly, let d represent a document vector with single term and phrase subvectors d_s and d_p. The simple innerproduct function (5) is used as the basic similarity function for a pair of subvectors, for example, q_s and d_s:

$$ip(q_s, d_s) = \sum_{i=1}^{k} q_{si} \cdot d_{si} \tag{5}$$

where k represents the length of subvector s, and q_{si} and d_{si} are the weights of the ith terms in the single term subvectors q_s and d_s.

For the single term subvectors to which the cosine normalization has been applied (see (3) above), the innerproduct function yields a similarity value equivalent to the cosine similarity function (Salton and Lesk 1971:163) applied to vectors to which the cosine normalization has not been applied.

The overall similarity value for vectors q and d is calculated as a weighted sum of the innerproduct similarity values calculated for the single term and phrase subvectors:

$$sim(q, d) = (c_s \cdot ip(q_s, d_s)) + (c_p \cdot ip(q_p, d_p)) \tag{6}$$

where c_s and c_p are weights applying to the single term and phrase subvectors, respectively. For the experiments reported on here, the value 1.0 has been used for both c_s and c_p.

With these weighting and similarity functions, the addition of phrase descriptors to document and query vectors has only a simple additive effect on the overall similarity between a document and query. That is, the partial similarity due to the single term subvector is not altered by the addition of phrase descriptors. The net effect of this strategy for weighting descriptors and calculating similarity values is that phrase descriptors can increase the similarity between a pair of vectors, but cannot reduce the partial similarity due to matches between descriptors in the single term subvectors of the query and document.

[14] For further discussion of composite vectors, see Fox (1983a, b).

REFERENCES

Aladesulu, O. S. 1985. Improvement of Automatic Indexing through Recognition of Semantically Equivalent Syntactically Different Phrases. Ph.D. Thesis, Ohio State University, Department of Information and Computer Science, Columbus, Ohio.

Baxendale, P. B. 1958. Machine-Made Index for Technical Literature – An Experiment. *IBM Journal of Research and Development* 2:354-361.

Baxendale, P. B. 1961. An Empirical Model for Machine Indexing. In: *Machine Indexing: Progress and Problems.* Third Institute for Information Storage and Retrieval, February 13-17, 1961. Center for Technology and Administration, School of Government and Public Administration, The American University, Washington, D.C.: 207-218.

Buckley, C. 1985. Implementation of the SMART Information Retrieval System. Technical Report TR85-686, Department of Computer Science, Ithaca, New York.

Croft, B. W. 1986. Boolean Queries and Term Dependencies in Probabilistic Retrieval Models. *Journal of the American Society for Information Science* 37(2):71-77.

Dillon, M. and A. S. Gray. 1983. FASIT: A Fully Automatic Syntactically Based Indexing System. *Journal of the American Society for Information Science* 34(2):99-108.

Doyle, L. B. 1961. Semantic Road Maps for Literature Searchers. *Journal of the Association for Computing Machinery* 8(4):553-578.

Doyle, L. B. 1962. Indexing and Abstracting by Association. *American Documentation* 13(4):378-390.

Earl, L. L. 1970. Experiments in Automatic Indexing and Extracting. *Information Storage and Retrieval* 6:313-334.

Earl, L. L. 1972. The Resolution of Syntactic Ambiguity in Automatic Language Processing. *Information Storage and Retrieval* 8(6):277-308.

Fagan, J. L. 1985. Using PLNLP for Content Analysis in Information Retrieval. Paper presented at the Symposium on PLNLP: The Programming Language for Natural Language Processing at the Annual Meeting of the Linguistic Society of America, Seattle, Washington, December 27-30, 1985.

Fagan, J. L. to appear. The Effectiveness of a Non-syntactic Approach to Phrase Indexing for Document Retrieval. *Journal of the American Society for Information Science.*

Fox, E. A. 1983a. Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts. Technical Report TR83-561, Department of Computer Science, Cornell University, Ithaca, New York.

Fox, E. A. 1983b. Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. Ph.D. Thesis, Cornell University, Department of Computer Science, Ithaca, New York.

Giuliano, V. E. and P. E. Jones. 1963. Linear Associative Information Retrieval. In: Paul W. Howerton and David C. Weeks, *Vistas in Information Handling, Volume I: The Augmentation of Man's Intellect by Machine.* Spartan Books, Washington, D.C.: 30-54.

Harper, D. J. and C. J. van Rijsbergen. 1978. An Evaluation of Feedback in Document Retrieval Using Cooccurrence Data. *Journal of Documentation* 34(3):189-206.

Heidorn, G. E. 1972. Natural Language Inputs to a Simulation Programming System. Technical Report NPS-55HD72101A, Naval Postgraduate School, Monterey, California.

Heidorn, G. E. 1975. Augmented Phrase Structure Grammars. In: R. Schank and B. L. Nash-Webber, *Theoretical Issues in Natural Language Processing: An Interdisciplinary Workshop in Computational Linguistics, Psychology, Linguistics, and Artificial Intelligence, 10-13 June 1975.* : 1-5.

Hillman, D. J. 1973. Customized User Services via Interactions with LEADERMART. *Information Storage and Retrieval* 9:587-596.

Hillman, D. J. and A. J. Kasarda. 1969. The LEADER Retrieval System. *AFIPS Proceedings* 34:447-455.

Jensen, K. 1986. PEG 1986: A Broad-Coverage Computational Syntax of English. Research Report, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

Klingbiel, P. H. 1973a. Machine-Aided Indexing of Technical Literature. *Information Storage and Retrieval* 9(2):79-84.

Klingbiel, P. H. 1973b. A Technique for Machine-Aided Indexing. *Information Storage and Retrieval* 9(9):477-494.

Lesk, M. E. 1969. Word-Word Associations in Document Retrieval Systems. *American Documentation* 20(1):27-38.

Metzler, D. P., T. Noreault, L. Richey, and B. Heidorn. 1984. Dependency Parsing for Information Retrieval. In: C. J. van Rijsbergen, *Research and Development in Information Retrieval: Proceedings of the Third Joint BCS and ACM Symposium, Kings College, Cambridge, 2-6 July 1984.* Cambridge University Press, Cambridge: 313-324.

Salton, G. 1966. Automatic Phrase Matching. In: D. G. Hays, *Readings in Automatic Language Processing.* American Elsevier Publishing Co., Inc., New York: 169-188.

Salton, G. 1968. *Automatic information storage and retrieval.* McGraw-Hill, New York.

Salton, G. 1986. Another Look at Automatic Text-Retrieval Systems. *Communications of the Association for Computing Machinery* 29(7):648-656.

Salton, G., C. Buckley, and C. T. Yu. 1983. An Evaluation of Term Dependence Models in Information Retrieval. In: Gerard Salton and Hans-Jochen Schneider, *Research and Development in Information Retrieval, Proceedings of the SIGIR/ACM Conference, Berlin, May 18-20, 1982.* Lecture Notes in Computer Science, 146. Springer-Verlag, Berlin: 151-173.

Salton, G. and M. E. Lesk. 1971. Computer Evaluation of Indexing and Text Processing. In: Gerard Salton, *The SMART Retrieval System -- Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey: 143-180.

Salton, G. and C. S. Yang. 1973. On the Specification of Term Values in Automatic Indexing. *Journal of Documentation* 29(4):351-372.

Salton, G., C. S. Yang, and C. T. Yu. 1975. A Theory of Term Importance in Automatic Text Analysis. *Journal of the American Society for Information Science* 26(1):33-44.

Smeaton, A. F. 1986. Incorporating Syntactic Information into a Document Retrieval Strategy: an Investigation. In: Fausto Rabitti, *Proceedings of the 1986 ACM Conference on Research and Development in Information Retrieval, Pisa, Italy, September 8-10, 1986.* Association for Computing Machinery, Baltimore, Maryland: 103-113.

Sparck Jones, K. 1972. A Statistical Interpretation of Term Specificity and Its Application to Retrieval. *Journal of Documentation* 28:11-20.

Sparck Jones, K. 1974. Automatic Indexing. *Journal of Documentation* 30(4):393-432.

Stiles, H. E. 1961. The Association Factor in Information Retrieval. *Journal of the Association for Computing Machinery* 8(2):271-279.

van Rijsbergen, C. J. 1977. A Theoretical Basis for the Use of Cooccurrence Data in Retrieval. *Journal of Documentation* 33:106-119.

Yu, C. T., C. Buckley, K. Lam, and G. Salton. 1983. A Generalized Term Dependence Model in Information Retrieval. Technical Report TR83-543, Department of Computer Science, Cornell University, Ithaca, New York.