Document Classification using Multiword Features

Ron Papka and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{papka,allan}@cs.umass.edu

Abstract

We investigate the use of multiword query features to improve the effectiveness of text-retrieval systems that accept natural-language queries. A relevance feedback process is explained that expands an initial query with single and multiword features. The multiword features are modelled as a set of words appearing within windows of varying sizes. Our experimental results suggest that windows of larger span yield improvements in retrieval over windows of smaller span. This result gives rise to a query contraction process that prunes 25% of the features in an expanded query with no loss in retrieval effectiveness.

1 Introduction

The following work investigates the representation for *queries* used in text-based information retrieval systems. The query representation described has applications in document filtering, routing, and clustering in addition to website searching. Our primary focus is the use of query features that represent concepts expressible in natural language by multiple words. We explore ways of obtaining useful multiple word features in a computationally tractable manner.

Text classification systems that perform various retrieval tasks are often implemented based on a model of word cooccurrence between a query representation and a potentially relevant document. Queries are provided by a user, or they are automatically generated with a query expansion process. The query representation for several commercial and research systems comprises words and weights of their relative importance. Users of web search-engines, such as Excite! and Infoseek, are currently entering very few words (without weights) per query [8]; however, the data from document retrieval research suggests that expanding queries with many words yields better results [3, 23, 12].

It becomes evident from examples that using queries comprising weighted independent single words, or their boolean combinations, limits retrieval effectiveness. Consider a user interested in information about *Harley Davidson Motorcycles.* Documents containing the word "Davidson" without the word "Harley" become unwanted relevant candidates using weighted independent words. Boolean combinations of words introduce ambiguity between the concept they are intended to represent, and other concepts that contain the same words. For example, a user interested in documents about *The World Bank* is not necessarily interested in retrieving documents about the merger that created *the world's largest bank.* In these examples, specifying the query as a phrase would assist in discriminating between relevant and non-relevant documents.

Much of the previous work that uses natural language phrases and other proximity constraints between multiple words suggests that "the closer a set of intersecting terms, the more likely they are to indicate relevance" [11]. Our experiments indicate that in the context of massive query expansion (queries with several hundred features), retrieval effectiveness improves by using query features implemented as sets of words that appear further separated within natural language text.

In the following section we discuss multiword features and how they are used in several document classification environments. A web-based query expansion process that uses a similar query representation to the work described here is presented in Section 3. In Section 4 we explain our massive query expansion process which is applicable to large-scale document filtering, routing and clustering tasks. Our query contraction process and ex-

To appear in *Proceedings of the Conference on Information and Knowledge Management*, November 1998.

perimental results are described in Section 5.

2 Related Work

A retrieval system that uses features exclusively mapped from single words has its drawbacks. Single word features are limited in the type of concept that can be represented by natural language. For example, the words "East" and "Coast" become a new concept when they are combined. Previous research in document classification extends the feature space by extracting natural language phrases and more general *multiword features.*

The utility of multiword features and their effects on retrieval has been mixed in the previous literature. Lewis [13] has documented some of the earlier work pertaining to representation and ambiguity issues arising from the use of phrases. In addition, TREC routing and filtering participants using multiword features do not appear to significantly outperform some participants that use only single word features [17].

These negative conclusions regarding multiword features are offset by positive results that have been reported. For example, Strzalkowski and Carballo [19] describe improvements in ad hoc retrieval using natural language phrases. Allan et al. [1] and Hawking et al. [11] report using proximity operators over sets of multiple words for improving filtering and ad hoc retrieval. Boolean features comprising multiple words have been used to improve precision by Hearst [12].

Results for other document classification problems show solutions using multiword features. Recent work by Carrick and Watters explained an application that matches news stories to photo captions using a frame-based approach focusing on proper noun phrases [6]. Riloff and Lehnert [16] also use multiword features as an integral part of a message understanding system that classifies news stories into pre-specified event frames.

There are several reported methods for extracting multiword features. N-gram models based on mutual information metrics are used to find sets of adjacent words that are likely to cooccur within sentences [2]. Part-of-speech tagging using pre-specified syntactic templates or more complex natural language parsing [9, 21] give rise to related multiple words comprising noun, verb, and prepositional phrases. In general, many of the previous techniques for extracting multiword features are based on finding phrases and multiple words that occur near each other within text; however, our approach suggests that modelling natural language concepts using words that may occur further apart improves retrieval.

3 Multiword Query Expansion

Queries comprising single word features appear to be sufficient for the typical internet search. A recent sample of roughly 500,000 queries from the Excite! search engine indicates that the average request is a query of 2.3 words [8]. A closer inspection of the corpus indicates that 95% of the searches do not make use of phrase query features even though the functionality for creating them is provided. We posit that simple and complex information requests benefit from representations that handle concepts that are expressible in natural language by more than one word. The major problem is to find multiword features that represent important concepts, automatically, and in a computationally tractable manner.

The integration of multiword features into the query representation implies additional support from the system's indexing processes, and requires new query language constructs for expressing concepts in natural language. One prevalent method for creating multiword features is to build system functionality that incorporates the proximity between words. Inquery [5, 1], for example, accepts queries containing ordered and unordered-window operators. They allow the user to specify an information request more precisely by imposing a locality constraint on groups of words.

Croft et al. tracked several thousand queries against the Thomas Congressional database and found that most queries were between 2 and 8 words[1]. In addition, they showed that internally translating the user's request into a query containing proximity operators provided better results than using the single words [7]. For example, the query "balanced budget amendment" would be translated to:

```
#WSUM( 1.0 1.0 balanced 1.0 budget 1.0 amendment
  90.0 #3( balanced budget amendment )
  45.0 #UW30( balanced budget amendment )
  90.0 #BAND( balanced budget amendment )
  20.0 #FIELD( TITLE #WSUM( 1.0
      1.0 balanced 1.0 budget 1.0 amendment
      20.0 #3( balanced budget amendment )
      10.0 #UW30( balanced budget amendment )
      1.0 #BAND( balanced budget amendment )))
  10.0 #PASSAGE200( balanced budget amendment ))
```

where #3, #UW30, #BAND, and #PASSAGE200 are proximity operators that can be calculated using term position information contained in the inverted file. The value preceding an operator represents its relative weight. Inquery's #WSUM operator is used as the comparison function between a query and the documents in the collection. This comparison function and a subset of these opera-

[1] A recent analysis indicated that the average Thomas query was 2.3 words as well.

tors are used in the experiments that follow and are described below.

The #3 operator requires that its arguments occur within 3 words of each other—e.g., `#3(nursing home)` allows up to two words between `nursing` and `home`. The #UW30 operator is similar, but looks for its arguments in an *unordered* window of 30 words—i.e., the words can occur in any order as long as they all occur within a set of 30 words. Multiple words appearing within a window of size 5 can be interpreted as words appearing within a phrase. A window of size 20 would represent a concept appearing within a sentence. A window of size 50 can be viewed as spanning a paragraph, and one of size 200 can be viewed as spanning adjacent paragraphs or a passage.

The #PASSAGE200 operator is like #UW200 operator except that it does not require that all of the words be present; it gives greater belief to 200-word passages that contain more of the words. Finally, #BAND is a Boolean "and" operator that looks for words that cooccur in the same document, regardless of the distance separating them.

4 Massive Query Expansion

4.1 Retrieval Representation

The massive query expansion process used in the experiments that follow is based on an implementation of the Inquery retrieval system [5]. For any comparison of document d from collection c to query q, we used the evaluation function of the #WSUM operator:

$$eval(q, d, c) = \frac{\sum_{i=1}^{|q|} w_i \cdot d_i}{\sum_{i=1}^{|q|} w_i} \qquad (1)$$

where $|q|$ is the number of features in the query, w_i is the weight of query feature q_i, and d_i is the *belief* that the feature's appearance in the document indicates relevance to the query.

Belief values are produced by Inquery's belief function, which uses a probabilistic inference network model [20]. In our implementation, the belief function is composed of a *feature frequency* component, *ff*, and an *inverse document frequency* component, *idf*. For any instance of document d and collection c:

$$d_i = belief(q_i, d, c) = 0.4 + 0.6 * ff * idf \qquad (2)$$

where

$$\begin{aligned} ff &= t/(t + 0.5 + 1.5 * \frac{dl}{avg\_dl}), \\ idf &= \frac{log(\frac{|c|+.5}{df})}{log(|c| + 1)}, \end{aligned}$$

t is the number of times feature q_i occurs in the document, *df* is the number of documents in which the feature appears in the collection, *dl* is the document's length, *avg_dl* is the average document length in the collection, and $|c|$ is the number of documents in the collection.

4.2 Expansion Process

The query expansion process used for this work is similar to and based upon those described in [3, 1]. The process has two steps: *feature selection* and *feature weight assignment.*

The primary focus of our experiments is on multiword feature selection. Our methodology for selecting expansion features begins with collecting the union of the stemmed words appearing in documents judged to be relevant and non-relevant. The top 50 words are sorted by the following metric and constraint:

$$\frac{r}{R} - \frac{n}{N} > 0 \qquad (3)$$

where r is the number of relevant documents in which the feature occurs, R is the total number of judged relevant documents, n is the number of non-relevant documents in which the feature occurs, and N is the total number of judged non-relevant documents. The metric is the percent of judged relevant documents minus the percent of judged non-relevant in which the feature occurs.

Expansion continues by adding multiword features. The word-pairs of each document are passed through proximity operators #1, #UW5, #UW20, #UW50, and the top 50 multiword features (based on Equation 3) for each window size are added to the query. In an effort to build multiword features comprising more than two words, the single and multiword features obtained from the process described thus far are passed through the #BAND operator, and the top 50 #BANDs are added to the query. Phrases are added at the beginning of the expansion processes using a technique described in [1].

In the following experiments query feature weights are assigned using a relevance feedback methodology similar to one originally developed by Rocchio [18]. The weight we assign to added query features is based on the *ff* component of the comparison model described above. For any added feature and set of judged documents, $w_i = 8 * ff_{rel} - 2 * ff_{nonrel}$, where ff_{rel} is the average *ff* component of the feature in relevant documents, and ff_{nonrel} is the average *ff* component in non-relevant documents. The weights for this Rocchio function were determined to work well empirically.

Expansion step	40 TREC-4 Queries Precision	 Improvement
Words only	15.2	
Words and top 50 #1	17.1	12.5%
Words and top 50 #UW5	17.6	15.8%
Words and top 50 #UW20	19.5	28.3%
Words and top 50 #UW50	20.3	33.6%
Words and top 50 #PHRASE	20.3	33.6%
Words and top 50 #BAND	24.0	57.9%

Table 1: Precision improvements on test set realized by expansion with multiword features.

Expansion step	40 TREC-4 Queries Precision	 Successive improvement
Words Only	15.2	
after also adding top 50 #PHRASE	20.3	33.6%
after also adding top 50 #1	21.6	6.4%
after also adding top 50 #UW5	23.2	7.4%
after also adding top 50 #UW20	25.1	8.2%
after also adding top 50 #UW50	25.8	2.8%
after also adding top 50 #BAND	27.4	6.2%

Table 2: Successive precision improvements on test set realized by massive query expansion.

5 Experiments

5.1 Data

Experiments were conducted on 50 natural-language information requests used for the routing track for TREC-4 [10]. The queries are a subset of TREC topics 3-191. The short description for each topic was stemmed, and then stopwords were removed to produce the initial query. Each query was subsequently expanded in several ways using the retrieval and expansion processes described in the section above.

The judged documents from Tipster volumes 1,2 and 3 were used for training, while a subset of the documents from the TREC-4 routing volume were used for testing. Volumes 1,2, and 3 contain 1,078,166 documents from the Associated Press(1988-90), Department of Energy abstracts, Federal Register(1988-9), San Jose Mercury News(1991), Wall Street Journal(1987-91), and Ziff-Davis Computer-select articles. The final test corpus contained 168,835 documents from the Federal Register(1994), IR Digest, News Groups, and Virtual Worlds. The average document contained 426 words in the training corpus and 408 words in the test corpus.

Given the judged documents available for volumes 1,2, and 3, on average 406 relevant documents and 1933 non-relevant documents were used to train each query. Several of the topics contained very few or no relevant documents in the test corpus, and we excluded 10 topics containing one or no relevant documents from testing. On average 59 relevant and 573 non-relevant documents were available to test the remaining 40 topics. Documents not judged to be relevant were considered non-relevant when evaluating retrieval results.

5.2 Evaluation Methodology

In the following experiments we evaluate the retrieval effectiveness of a query by its ability to generalize to new documents. In order to avoid threshold optimization, we show *non-interpolated average precision* and evaluate a query's ability to rank, in order of relevance, the entire test set of documents.

This metric can be explained as follows: assume there is a ranked list of documents sorted by $eval(q,d,c)$ — as described by Equation 1 — and that

a = correctly classified relevant documents,
b = incorrectly classified non-relevant documents,
c = incorrectly classified relevant documents, and
d = correctly classified non-relevant documents;

then, $Recall = \frac{a}{a+c}$, and $Precision = \frac{a}{a+b}$.

The final valued is then calculate by averaging *precision* at every point that *recall* increases (the appearance of a known relevant document) in the ranked list. We test significant retrieval improvements using a two-tailed sign test with $\alpha = 0.05$ as the coefficient.

	40 TREC-4 Queries			
	Average Number of Features		Ratio	Recall
Expansion step	REL	NON-REL	R : NR	at 1000
Words Only	23.1	15.0	1.540	0.6508
after also adding phrases	31.6	20.1	1.572	0.7550
after also adding top 50 #1	36.8	21.6	1.703	0.7916
after also adding top 50 #UW5	43.6	23.2	1.879	0.8140
after also adding top 50 #UW20	51.6	25.2	2.048	0.8266
after also adding top 50 #UW50	60.7	27.6	2.199	0.8342
after also adding top 50 #BAND	72.1	31.4	2.296	0.8646

Table 3: Average number of features cooccurring in the query and judged documents in the test set.

Window size	#1	#UW5	#UW20	#UW50
#1	-	8.1%	3.5%	2.5%
#UW5	-	-	17.7%	12.4%
#UW20	-	-	-	27.2%
#UW50	-	-	-	-

Table 4: Average percent of overlapping identical word-pairs occurring in different window sizes.

5.3 Results

In the context of massive query expansion using single and multiword query features, retrieval effectiveness improved using pairs of words that are further apart in text. Instead of a relevance feedback methodology that expands queries with representations of multiword concepts with near proximity, we are finding significantly better retrieval effectiveness using query representations modelling far proximity.

5.3.1 Query Expansion

The results in table 1 illustrate the effects of expanding queries containing single word features with the top 50 multiword features for each window size. This process resulted in queries with an average of 110 features. The data indicate that expansion with the top 50 multiword features represented by larger window sizes obtained higher precision than expansion using smaller window sizes. Expanding queries with #1s gives rise to a 12.5% improvement in precision on average. However, this improvement over single word features is far less than the 33.6% improvement realized by using the best #UW50s or the 57.9% improvement using the best #BANDs of single and multiword features. [2]

We tested our massive query expansion process (described in Section 4.2) which expanded the initial single word query with the best 50 multiword features from all the fixed-size window operators from Table 1. This process resulted in queries with an average of 321 features.

The data in Table 2 indicate that precision improves by expanding single word queries with multiword proximity features. Furthermore, precision continues to improve as features using window operators of greater span are added to the query. For example, after the initial query of words is expanded with phrases, #1s, and #UW5s (row 4 in the table), a significant increase in precision is obtained by adding the top 50 #UW20s (row 5). Each iteration of query expansion provides a significant improvement over the previous one except for the step that expands the query with the top 50 #UW50s. These results suggest that more features are better than fewer features. The data support the findings of Buckley et al. [3], where single word features were used. In addition, these results suggest that the positive effects of massive query expansion extend to multiword features.

In an effort to understand this phenomenon, we examined the query features and their cooccurrence in relevant versus non-relevant documents. Table 3 shows the average number of features that cooccur in the expanded query and judged documents. As the query is expanded the number of features that cooccur in the documents increase; however, on average, the growth rate of cooccurrence in the relevant documents (column 2 in the table) exceeds the growth rate of cooccurrence in the judged non-relevant documents (column 3) in the test set. This implies that the query's *cooccurrence ratio* between relevant and non-relevant documents is also increasing (column 4).

The last column of Table 3 shows recall in the

[2] The #UW50s and #PHRASEs have similar performance improvements. The #PHRASE operator is implemented using a non-linear combination of near and far proximity operators. The higher the frequency of the operator's words within the document, the more likely far proximity is modelled. Since the majority of documents in the test corpus are long Federal Register documents, it is most likely the case that word frequencies were relatively high and far proximity was used; however, this was not examined further.

Contraction step	40 TREC-4 Queries Precision
Expanded query	25.8
Expanded query, no dups	25.4
#1 replaced with #UW5	25.4
#UW5 replaced with #UW20	25.7
#UW20 replaced with #UW50	26.1

Table 5: Breakdown of precision improvements in test set.

	40 TREC-4 Queries		
	Average Number of Features		Ratio
Contraction step	REL	NON-REL	R : NR
Expanded query, no dups	50.1	24.8	2.020
#1 replaced with #UW5	50.7	25.1	2.020
#UW5 replaced with #UW20	53.0	26.2	2.023
#UW20 replaced with #UW50	55.5	27.3	2.033

Table 6: Average number of features cooccurring in the query and judged documents in the test set.

top 1000 ranked documents increasing consistently. The data indicate that the features with which we continue to expand our query are enhancing recall and improving precision. The precision improvement realized during training generalized to the test set, and is mostly attributed to the feature selection criterion described by Equation 3.

We tested approximately 60 combinations of the expansion order for each size window to ensure that the above results were not dependent on the order of expansion, and the following consistency appeared in the data:

> After any expansion step of adding 50 multiword features, queries containing proximity operators spanning bigger windows had better retrieval effectiveness than similarly sized queries using operators spanning smaller windows.

For example, we started with queries consisting of words, #PHRASEs, and #1s. When the query was expanded with the 50 #UW5 concepts, average precision rose. But it rose more when the 50 #UW20s were added instead, and still more when the #UW50s were used. This pattern repeated itself uniformly throughout all combinations we tried.

In retrospect it becomes evident that the weight assignment strategy for added features already exhibited a preference for bigger windows. The weights assigned to word-pairs appearing in different sized windows were greater for bigger windows than smaller ones 85% of the time. The definition of the unordered-window insures that the feature frequency of a bigger window is no less than the frequency for a smaller window using the same set of words. In addition, our data suggest that features from bigger window operators are, on average, better discriminators of relevant documents than features from smaller ones. It appears that the frequency statistics for smaller windows are not as statistically useful to our retrieval model which is based on feature frequency and inverse document frequency. The insufficient occurrence of natural language phrases has been concluded as a possible cause for no improvement using near-proximity multiword features [13].

5.3.2 Query Contraction Process

Reducing the number of features used to represent a query may have a positive impact on the computation time needed to process a query as well as retrieval effectiveness. Pruning irrelevant single word features after massive query expansion has been reported using machine learning approaches by Lewis et al. [14] and Papka et al. [15].

The contraction process we used to produce the final queries is based on the observation that several of the features using the unordered-window operators could be merged into one operator. By manual inspection, it was evident that the expansion process produced the same pairs of words for more than one unordered-window size. For example, 17.7% of the word-pairs appeared as both #UW5s and #UW20s. Table 4 shows the breakdown of the overlapping identical word-pairs between window sizes. In the average expanded query 37% of the word-pairs appeared as duplicates using different window sizes. This data, in conjunction with the hypothesis that bigger windows are better, led to the following query contraction process:

1. Use the feature selection and weight assignment strategy described above to build a massively expanded query.
2. Identify overlapping proximity operators that

contain identical sets of words.

3. Replace overlapping operators with one multiword feature modelling the largest size unordered-window.

4. The merged multiword feature weight is the sum of the feature weights identified in step 2.

Table 5 illustrates the breakdown of the effects of successively replacing smaller window operators with successively larger window operators. The first row of the table shows precision for the expanded queries which contain duplicates. When the duplicates are removed (row 2), the resulting queries have worse precision on average. When #UW5 operators are replaced with #UW20 operators using the same words, precision begins to increase (row 4). Precision continues to improve when #UW20s are replaced with #UW50s. The final set of queries contain single word features and #UW50s (row 5).

The contraction process produced a final query averaging 218 features, instead of the 289 features from the original expanded query [3]. Removing duplications and combining features into larger window sizes results in a set of queries with 25% fewer features than the set of massively expanded queries. In addition, the final smaller queries exhibit an improvement in precision on the test set.

We found the best results by setting the "largest size unordered-window" in step 3 to one that spans over 50 words. Using #UW200 operators and #BANDs yielded the same precision improvement as using #UW50 operators. This implies that the improvement tails-off after a certain span.

We analyzed the contraction process with the approach used for the expansion process. Table 6 shows the average number of features that cooccur in the contracted query and judged documents after each iteration of the contraction process. The data suggest that when larger windows are used during the contraction process, precision increases as the cooccurrence ratio increases.

The data in Tables 3 and 6 illustrate that an increase in feature cooccurrence ratio is positively correlated with improved precision during both expansion and contraction. It is evident from the tables that the differences in ratios are not as dramatic for contraction as for expansion; however, we believe the cause for equivalent retrieval effectiveness between the queries resulting from the contraction process and the expansion process is because bigger windows of words ultimately discriminate relevant documents better than smaller ones.

6 Conclusion

In the context of query expansion, methods for representing and selecting good multiword features in a computationally tractable manner are complex. In the past, the utility of such features has been met with mixed results. Several of the previous expansion approaches incorporating phrases and other multiword features have focused on modelling close proximity for multiple words. Our experiments indicate that better precision results when queries are expanded with features that are modelled as a set of words that appear further separated within natural language text.

We have shown automatic techniques applicable to document classification tasks where the expansion of a query with multiword features led to significant improvements in retrieval effectiveness over queries using independent single word features. Results from our massive query expansion process suggest that when modelling multiword features using proximity constraints, bigger unordered-windows yield greater precision gains than smaller windows. As features represented by windows of greater span are added to queries, precision improves over similar queries using smaller windows. The benefit of the larger window is apparently due to their more frequent appearance in relevant documents versus non-relevant documents. Our hypothesis was tested in a query contraction process that merged overlapping proximity features to produce queries that were 25% smaller than the original with similar retrieval effectiveness.

We examined the effects of query expansion with multiword features and found a correlation between increasing cooccurrence ratios and improved precision. Our analysis has addressed the impact of feature cooccurrence, but we have yet to address the effects of feature weight assignment. We anticipate that feature weight learning approaches applied to single word features [4, 14, 15] will further improve retrieval effectiveness for queries with multiword features.

We are continuing to investigate the value of phrases and multiword features for several document classification tasks. Our belief is that query expansion approaches are limited when only independent single word features are used to represent queries.

[3] The #BAND operator from the expansion process described earlier contains mixes of features and a different document weighting function than the other proximity operators. For that reason, the baseline set of queries used to describe the contraction process excluded the #BANDs from evaluation.

Acknowledgments

Special thanks to Don Byrd, Jay Ponte, and Mark Sanderson for stimulating discussions and calculations of the Excite! and Thomas query corpora.

This material is based on work supported by the National Science Foundation, Library of Congress, and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors' and do not necessarily reflect those of the sponsor.

References

[1] J. Allan, L. Ballesteros, J. Callan, W.B. Croft, and Z. Lu, "Recent Experiments with Inquery," *Proceedings of TREC-4*, 49-64, 1996.

[2] P. F. Brown, J. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin, "A Statistical Approach to Machine Translation," *Computational Linguistics*, 16(2): 79-85, 1990.

[3] C. Buckley, G. Salton, and J. Allan, " The Effect of Adding Relevance Information in a Relevance Feedback Environment," *Proceedings of SIGIR*, 49-58, 1993.

[4] C. Buckley and G. Salton, "Optimization of Relevance Feedback Weights," *Proceedings of SIGIR*, 351-357, 1995.

[5] J. P. Callan, W.B. Croft, and S.M. Harding, "The INQUERY Retrieval System," *Database and Expert Systems Applications: Proceedings of the International Conference in Valencia Spain*, A.M. Tjoa and I. Ramos eds., Springer Verlag, New York, 1992.

[6] C. Carrick, and C. Watters, "Automatic Association of New Items," *Information Processing & Management*, 33(5):615-632, 1997.

[7] W.B. Croft, R Cook, and D. Wilder, "Providing Government Information on the Internet: Experiences with THOMAS," *Proceedings of Digital Libraries Conference*, 19-24, 1995.

[8] D. Cutting, "Industry Panel Discussion," SIGIR, 1997. (A sample of Excite! queries from September 16, 1997 became available after this discussion.)

[9] J.L. Fagan, *A Comparison of Syntactic and Non-Syntactic Methods*, Ph.D. Thesis, Department of Computer Science, Cornell University, 1987.

[10] D. Harman, *Proceedings of Text REtrieval Conferences (TREC)*, 1993-1997.

[11] D. Hawking, and P. Thistlewaite, "Proximity Operators - So Near And Yet So Far," *Proceedings of TREC-4*, 131-144, 1996.

[12] M. A. Hearst, "Improving Full-Text Precision on Short Queries using Simple Constraints," *Proceedings of SDAIR*, 1996.

[13] D.D. Lewis, *Representations and Learning In Information Retrieval*, Ph.D. Thesis, Department of Computer and Information Science, University of Massachussetts, 1991.

[14] D. Lewis, R. Schapire, J. Callan, and R. Papka, "Training Algorithms for Linear Text Classifiers," *Proceedings of SIGIR*, 298-306, 1996.

[15] R. Papka, J. Callan, and A. Barto, "Text-Based Information Retrieval using Exponentiated Gradient Descent," *Proceedings of Neural Information Processing Systems Conference*, 3-9, 1996.

[16] E. Riloff and W. Lehnert, "Information Extraction as a Basis for High-Precision Text Classification," *ACM Transactions on Information Systems*, 12(3): 296-333, 1994.

[17] S.E. Robertson, S. Walker, M.M. Beaulieu, M. Gatford, and A. Payne, "Okapi at TREC-4," *Proceedings of TREC-4*, 73-86, 1996.

[18] J.J. Rocchio, "Relevance Feedback in Information Retrieval," in *The Smart System - Experiments in Automatic Document Processing*, 313-323, Englewood Cliffs, NJ: Prentice Hall Inc. 1971.

[19] T. Strzalkowski and J. Perez Carballo, "Natural Language Information Retrieval: TREC-4 Report," *Proceedings of TREC-4*, 245-258, 1996.

[20] H. Turtle and and W.B. Croft, "A Comparison of Text Retrieval Models," *Computer Journal*, 35(3): 279-290, 1992.

[21] E. Tzoukermann, J.L. Klavans, and C. Jacquemin, "Effective Use of Natural Language Processing Techniques for Automatic Conflation of Multi-Word Terms: The Role of Derivational Morphology, Part of Speech Tagging, and Shallow Parsing," *Proceedings of SIGIR*, 148-155, 1997.

[22] C.J. van Rijsbergen, *Information Retrieval, 2ed.*, Butterworths, Massachusetts, 1979.

[23] E. M. Voorhees, "Query Expansion using Lexical-Semantic Relations," *Proceedings of SIGIR*, 311-317, 1994.