Why Bigger Windows Are Better Than Smaller Ones

Ron Papka and James Allan
*Department of Computer Science
University of Massachusetts
Amherst, MA 01003
papka@cs.umass.edu, allan@cs.umass.edu

Abstract

We investigate the use of multi-term query concepts to improve the performance of text-retrieval systems that accept "natural-language" queries. A relevance feedback process is explained that massively expands an initial query with single and multi-term concepts. The multi-term concepts are modelled as a set of words appearing within windows of varying sizes. Experimental results suggest that windows of larger size yield improvements in average precision. The reason for this improvement is explored. A window size relaxation process that yields a significant reduction in expanded query size with no performance loss is also described.

1 Introduction

The general intuition about multi-term concepts is that "the closer a set of intersecting terms, the more likely they are to indicate relevance" [4, 5]. Our experiments indicate that, contrary to intuition, in the context of relevance feedback, performance gains are obtained by using query concepts that are modelled as pairs of terms that appear further separated within natural language text.

Since these results are counter-intuitive, they deserve further exploration, and upon closer examination of the $tf.idf$ document representation commonly used in IR literature, it appears that a bias exists in the context of massive query expansion (queries with several hundred to a few thousand concepts). This bias favors tf over idf, and results in higher scores for documents containing more of the concepts in the query. As concepts that are represented by relatively larger windows are added to queries, performance improves, apparently due to the relative rate at which these concepts co-occur in relevant documents versus non-relevant documents.

We discuss multi-term concepts and how they are used in general information retrieval environments. We explain massive query expansion experiments used in a routing environment, which attempt to reveal the change in document representation caused by the expansion of the set of query concepts. From these results we conclude that bigger windows appear better than smaller ones.

*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 author and do not necessarily reflect those of the sponsor.

2 Multi-term Concepts

One prevalent method for creating multi-term concepts in IR systems is to build functionality that incorporates the proximity between terms. INQUERY [8, 11], for example, accepts queries containing ordered and unordered window operators. This functionality allows the user to specify an information request more precisely by imposing a locality constraint on groups of words.

Proximity can assist in discriminating between relevant and non-relevant documents. Consider a news brief document that contains excerpts pertaining to home sales and the current status of the nursing industry. If a user wanted to retrieve documents about "nursing homes", the news brief just described would be a relevant candidate; however, if the system treated "nursing home" as one query concept, where the word *nursing* must immediately precede the word *home*, the news brief would be non-relevant.

Researchers tracking several thousand queries against the Thomas Congressional database found that most queries were between 2 and 8 words [15]. In addition, they showed that internally translating the user's request into a query containing proximity operators provided better results than using the single terms. For example, the query "balanced budget amendment" would be translated to:

```
#WSUM( 1.0 1.0 balance 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 balance 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 #PARSUM200( balanced budget amendment ))
```

where #3, #UW30, #BAND, and #PARSUM200 are proximity operators that can be calculated using term position information contained in the inverted file. The value preceding a query concept represents its relative weighting in belief calculations using a sum operator #WSUM, which calculates the weighted average belief of the concepts in the query co-occurring in the document. The belief function is described in more detail below.

The proximity operator #3 uses an ordered list of two or more words, and retrieves for each document the number of times the words appear in order within three positions of each other. For example, the syntax #3(nursing home), would count the number of times "home" followed the word "nursing" with at most two words in between.

The unordered-window operator, #UW30 for example, gives rise to concepts that are modelled as a group of words appearing together within a stretch of 30 words. Two words appearing within a window of size 5 can be interpreted as a concept represented by the words appearing within a sentence. A window of size 20 would represent a concept appearing within adjacent sentences, a window of size 50 can be thought of as a paragraph, and one of size 250 can be thought of as a "passage" [6].

The #BAND (boolean AND) operator is the generalization of the unordered window operator to the entire document. The concepts that are processed by the #BAND operator can appear anywhere in the document, and at any distance from one another, in order for the concept being represented to be considered present in the document.

Many text-based IR requests are single-term concept based. This is sufficient for narrow bandwidth internet searching; however, more complex requests, such as those received by routing applications, will require query constructors that handle concepts that are expressible in natural language by more than one word.

3 Retrieval Process

Text-based information retrieval systems allow the user to pose a query to a collection or a stream of documents. When a query q is presented to a collection c, each document $d \in c$ is examined and assigned a value relative to how well d satisfies the request posed by q. For any instance of the triple $< q, d, c >$, the system determines an evaluation value attributed to d using the function $eval(q, d, c)$. For example, the evaluation function used by the #WSUM operator is the weighted average:

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

where w_i is the relative weight of a query concept q_i, and d_i is the *belief* that the concept indicates relevance to the query.

A ranking of documents based on query q is achieved by sorting all documents in a collection by evaluation value. Binary classification is achieved by determining a threshold θ such that for class R, $eval(q, d, c) \geq \theta \rightarrow d \in R$, and $eval(q, d, c) < \theta \rightarrow d \in \bar{R}$, so that R is the set of documents from the collection that are classified as relevant to the query, and $\bar{R}$ is the set classified as non-relevant[2, 3].

The document representation for the expansion process described below is a set of belief values corresponding to each concept specified in a query. Belief values are produced by INQUERY's belief function which is composed of a *term frequency* component, tf, and an *inverse document* frequency component, idf, described in [11, 14]. The tf component causes the belief in a document to increase as a query concept's occurrence in the document increases, and the idf component causes the belief in a document to decrease as the number of documents in the collection in which the concept occurs increases. The belief function is invoked for each query concept q_i to create a document representation containing a set of beliefs. For any instance of document d and collection c:

$$d_i = belief(q_i, d, c) = 0.4 + 0.6 * tf * idf \tag{2}$$

where $tf = t/(t + 0.5 + 1.5 * \frac{dl}{avg\_dl})$, $idf = \frac{log(\frac{C+.5}{df})}{log(C+1)}$, t is the concept's frequency in the document, df is the number of documents in which the concept 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.

Figure 1 shows the behavior of the belief function for changes in t and df using an average length document. Concepts that do not appear in a document receive a default belief of 0.4, while concepts appearing seldom in the collection, and many times in a document receive the highest belief (1.0). The belief function is also applied to concepts contained in the #BAND operator where the beliefs of the individual concepts are multiplied if all concepts appear in a document. Otherwise, the belief of the #BAND assumes the default belief of $0.4^{number\_of\_concepts}$.

4 Query Expansion Process

The query expansion processes used for this work is similar to those described in [1, 11, 2, 3] which exhibit good performance in routing environments[1, 13] using a two step methodology of *concept*

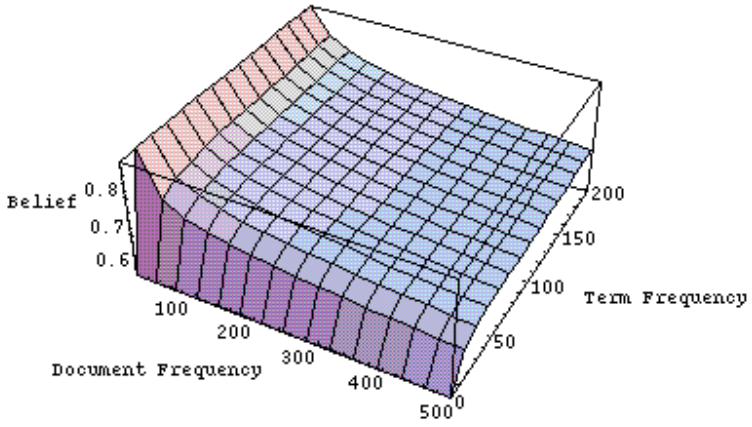

Figure 1: Belief function varying term (concept) and document frequencies

selection and *weight assignment*.

4.1 Concept selection

Our methodology for selecting expansion concepts begins with collecting the union of the terms appearing in documents judged to be relevant. The top 50 terms sorted by the metric and constraint: $\frac{r}{R} - \frac{nr}{NR} > 0$, are added to the query. The metric is the ratio of judged relevant documents minus the ratio of judged non-relevant documents within which the term appears. The $O(n^2)$ term-pairs are passed through proximity constructors of #1, #UW5, #UW20, #UW50, and the top 50 concepts (based on the metric above) for each window size are added to the query. Finally, the pairs of concepts from the expanded query 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 lexographical technique described in [11].

The focus of our experiments is on multi-term concept selection. Buckley et al.[1] show evidence that, in the context of relevance feedback, increasing the number of expansion terms improves performance. (Performance tends to level off after at a point of 200 to 500 terms.) The optimal number of concepts by which to expand a query is most likely domain-dependent and could therefore be learned, but that would require additional experimentation. Unfortunately, early results suggest that query-specific tailoring of these numbers may not be effective.[16, 17]

4.2 Weight Assignment

A step popular in relevance feedback methodology is to assign query term weights based on a closed-form function originally developed by Rocchio[9], and has been improved upon in [10, 7, 11]. The

weight assigned to a concept added to an expanded query is $8 * tf_{rel} - 2 * tf_{nonrel}$, where tf_{rel} is the average tf component of the concept in relevant documents, and tf_{nonrel} is the average tf component in non-relevant documents. Recent research[7, 11, 2] indicates that iterative techniques can be used to improve these weights.

5 Experiments

Experiments were conducted on 50 natural-language information requests used for the routing track for TREC-4[1]. The information requests were stopped and stemmed to produce an initial query, and subsequently expanded in several ways using the retrieval and expansion processes described above.

5.1 Data

The 50 queries used were the topics developed for the routing track at Text Retrieval Conferences (TREC)[13]. The queries are from TREC-4, topics 3-191.

The judged documents from Tipster volumes 1,2 and 3 were used for training, while 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 routing volume contains 329,780 different documents from similar sources. 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.

5.2 Evaluation

In the experiments that follow, Simple Average Precision (SAP) was used to evaluate ranking performance. This metric can be explained as follows: assume there exists a set of documents sorted by $eval(q, d, c)$ — as described in Section 3—and that

a = number of relevant documents classified as relevant,
b = number of non-relevant documents classified as relevant,
c = number of relevant documents classified as non-relevant, and
d = number of non-relevant documents classified as non-relevant;

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

Precision and *recall* can be calculated at any rank—i.e., at any cut-off point in the sorted list of documents. *Simple Average Precision* is the mean precision measured from all cut-off points associated with documents judged to be relevant to the query.

6 Results

Contrary to the intuition that "[t]he closer a set of intersecting terms, the more likely they are to indicate relevance,"[4] we find significantly more of the queries for TREC-4 routing show the following behavior:

> In the context of massive query expansion using single-term and multi-term query concepts, recall-precision based performance metrics improve as the proximity of pairs of terms is relaxed.

Instead of a relevance feedback methodology that expands queries with representations of multi-term concepts with near proximity, we are finding significantly better performance in using representations modelling far proximity.

	Topics 3-191: 50 Queries	
Expansion type	SAP	Improvement over Terms
Terms	20.7	
Terms and top 50 #1	24.6	18.8%
Terms and top 50 #UW5	26.3	27.1%
Terms and top 50 #PHRASE	26.8	29.5%
Terms and top 50 #UW20	28.9	39.6%
Terms and top 50 #UW50	29.3	42.0%
Terms and top 50 #BAND	33.6	62.0%

Table 1: Effects of adding query concepts using proximity operators.

As Table 1 shows, expanding queries with #PHRASEs (which use both near-proximity operators and within-same-document operators) gives rise to a 29.5% improvement in SAP, as the hypothesis would suggest. However, this improvement is far less significant than the 42% improvement realized by #UW50s or the 62% improvement using #BANDs of concepts.[1] The results in Table 1 are significant based on two-tailed sign tests with $\alpha = .05$ confidence. They suggest that expansion using concepts with bigger windows is better.

But, why are bigger windows better? Before answering that question, we first looked at the cause for the significant improvements seen by adding more query concepts. To our surprise, the reason bigger windows are better is related to the reason the query expansion process works well in the relevance feedback environment described above.

6.1 Expansion Process

An experiment studying the effects of the expansion process confirms the findings by Buckley et al[1]: adding query concepts that appear in the set of judged relevant documents gives rise to performance improvements in the test set. Furthermore, their results generalize to multi-term proximity concepts.

Table 2 shows the performance improvements in the test set realized by the successive iterations of query expansion. Each iteration provides a significant improvement (as measured by a two-tailed sign test with $\alpha = .05$) over the previous one except for the step that expands the query with the top 50 #UW50. The final queries have on average 350 concepts.

[1] The #BAND queries were constructed from pairs of concepts—terms or pairs of terms—that were highly correlated with relevant documents, so they include a selection of concepts of varying window sizes. Their broad coverage may account for some of the large improvement shown by that operator.

Expansion type	Topics 3-191: 50 Queries SAP	Successive improvement
Terms Only	20.7	
after adding phrases	26.8	29.5%
after adding top 50 #1	28.9	7.8%
after adding top 50 #UW5	31.3	8.3%
after adding top 50 #UW20	34.4	9.9%
after adding top 50 #UW50	35.9	4.4%
after adding top 50 #BAND	38.2	6.4%

Table 2: Effects of query expansion on Simple Average Precision.

Table 3 shows the average number of concepts that co-occur in the expanded query and relevant and non-relevant documents. The growth rate of query concepts co-occurring in the average relevant document exceeds the growth rate of co-occurring concepts in the average judged non-relevant document in the test set. Hence, the concepts with which we continue to expand our query are enhancing recall without affecting precision. (The final column of Table 3 shows recall in the top 1000 retrieved documents increasing consistently.) This improvement in precision *and* recall is due to the belief scores for relevant documents increasing, while those in non-relevant documents do not increase enough to offset the performance gain we experience in Table 2. The belief function depicted in Figure 1 ensures that if a concept appears in the document, overall belief will increase; otherwise overall belief decreases towards a constant default belief. Therefore, we experience an increase in average precision at these performance levels.

Expansion type	Topics 3-191: 50 Queries Average Number of Concepts REL	NON-REL	Recall at 1000
Terms Only	21.6	17.5	0.6044
after adding phrases	27.9	22.1	0.7289
after adding top 50 #1	31.6	23.7	0.7457
after adding top 50 #UW5	36.7	25.6	0.7639
after adding top 50 #UW20	44.3	28.4	0.7799
after adding top 50 #UW50	53.2	31.9	0.7876
after adding top 50 #BAND	68.7	37.5	0.8182

Table 3: Average number of concepts co-occurring in the query and documents in the test set.

6.2 Relaxing Window Size

Approximately 60 combinations of the expansion order for concept types were tested 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 concepts, queries containing concepts using bigger windows had better performance than similarly sized queries using smaller windows.

For example, we started with queries consisting of terms, phrases, and #1s. When the query was expanded with the 50 #UW5 concepts, average precision rose. But it rose more when the 50 #UW20's were added instead, and still more when the #UW50's were used. This pattern repeated itself uniformly throughout all combinations we tried.

In retrospect, it becomes clear that the Rocchio weight assignments for added concepts already showed a preference for bigger windows. The weights assigned to terms pairs appearing in different sized windows were greater for bigger windows than smaller ones 85% of the time. In general, 30% of unique pairs of terms appeared as duplicates in the top 50 concepts selected for more than one window size. Table 4 shows the breakdown of the overlap between window sizes. Learning new weights using the Dynamic Feedback Optimization technique[7, 11] showed a slight tendency to increase the weights of smaller windows, but even after the weights were adjusted, the bigger window received the higher weight 70% of the time.

Query type	#1	#UW5	#UW20	#UW50
#1	-	7.6%	3.5%	2.6%
#UW5	7.6%	-	18.0%	12.8%
#UW20	3.5%	18.0%	-	28.0%
#UW50	2.6%	12.8%	28.0%	-

Table 4: Percent of overlapping term pairs between window sizes.

The purpose of the relaxation process was to look at the effects of changing window size on document representations. First, we removed duplicate pairs of terms and *rolled* their weights into the larger window leaving total query weight constant. For each iteration, we modified the query by widening the smallest windows to the next largest size, while leaving the concept's query weight unchanged.

Table 5 shows the changes in performance during each iteration of the relaxation process. Removing duplications and relaxing window sizes results in a set of queries with 30% fewer concepts than the expanded query set (an average of 218 concepts rather than the original 311). The expanded query set is significantly better than the query set without duplications; however, the final query set produced by the relaxation process is better for 56% of the queries than the original query after expansion and before relaxation.

Relaxation type	Topics 3-191: 50 Queries SAP
Expanded query	35.9
Expanded query, no dups	35.0
#1 Relax #UW5	35.2
#UW5 Relax #UW20	35.8
#UW20 Relax #UW50	36.2

Table 5: Change in performance due to relaxation processes.

Table 6 shows that during the relaxation process, changes take place in the document representation. The bigger window modelling multi-term concepts gives rise to more co-occurrences in more documents than the same set of terms in a smaller window. The number of query concept co-occurrences in non-relevant documents is not enough to offset the performance gains brought about by an increase in overall-belief for relatively more relevant documents. The bias of the belief function insures that at the experienced performance levels, overall belief in a document will increase for each additional concept that co-occurs in the query.

The results show that expanding the window size in which pairs of terms are expected to appear causes an increase in effectiveness. An expanded window size generally increases the number of occurrences of the term pair (they will occur more often within 10 of each other than they will occur adjacently), increasing both the number of documents containing the concept (lowering idf) and the number of times the concept occurs in a document (increasing t). At low t-values, increases in the tf component have a stronger impact on the belief score used by INQUERY than do decreases in the idf, so belief scores stay approximately constant despite the addition of more occurrences.

Relaxation type	Topics 3-191: 50 Queries Average Number of Concepts REL	NON-REL
Expanded query, no dups	41.3	27.5
#1 Relax #UW5	42.4	28.1
#UW5 Relax #UW20	45.5	29.8
#UW20 Relax #UW50	48.5	31.5

Table 6: Average number of concepts co-occurring in the query and documents in the test set.

When the concepts are "relaxed" (to a wider window size), more of the concepts in a query match documents in the collection. Fortunately, the increase in the number matching grows faster in relevant than in non-relevant documents, so the net result is an improvement in effectiveness.

7 Conclusion

Experimental results suggest that when modelling concepts using proximity constraints represented by multi-term window operators, bigger windows contribute to greater improvements in average precision than do smaller windows. This hypothesis is tested in a massive query expansion step using pairs of concepts within the #BAND operator. This expansion step led to significantly improved performance.

Our hypothesis is also tested in a relaxation process that eventually produces a query that is 30% smaller than the original, with the same retrieval performance. In an effort to understand this phenomenon, we also studied our expansion process, and realized that one of the reasons that it works is the same reason that bigger windows are better than smaller ones.

We are continuing to investigate the value of wider windows as part of an exploration of concept identification during relevance feedback. Our belief is that relevance feedback approaches are limited by the nature of the concepts currently used. Simple reweighting of mediocre concepts will not be sufficient to transcend the current levels of effectiveness found in research systems.

References

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

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

[3] R. Papka, J. Callan, and A. Barto, "Text-Based Information Retrieval using EG", Proceedings of Neural Information Processing Systems Conference, 1996. Forthcoming.

[4] D. Hawking, and P. Thistlewaite, "Proximity Operators - So Near And Yet So Far", Proceedings of TREC-4 (1995), pp. 131-144.

[5] C. Clarke and G. Cormak, "Interactive Substring Retrieval: MultiText Experiments for TREC-5", Proceedings of TREC-5 (1996). Forthcoming.

[6] J. Allan, "Relevance Feedback With Too Much Data," Proceedings of SIGIR 1995, pp. 337-343.

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

[8] 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.

[9] J.J. Rocchio, "Relevance Feedback in Information Retrieval in The Smart System - Experiments in Automatic document processing", pp. 313-323. Englewood Cliffs, NJ: Prentice Hall Inc., 1971.

[10] G. Salton, and C. Buckley, "Improving Retrieval Performance by Relevance Feedback", Journal of The American Society For Information Science 41(4), 1990.

[11] J. Allan, L. Ballesteros, J. Callan, W.B. Croft, and Z. Lu, "Recent Experiments with Inquery", Proceedings of TREC-4 (1995), pp. 49-64.

[12] J. Zobel and A. Mofit, "Similarity Measures Explored," Department of Computer Science, RMIT Technical Report CITRI/TR-95-3, 1995.

[13] D. Harman, Proceedings of Text REtrieval Conferences (TREC), 1993-1996.

[14] S.E. Robertson, W. Walker, S. Jones, M.M. Hancock-Beaulieu, and M.Gatford, "Okapi at TREC-3", Proceedings of TREC-3 (1994), pp. 109-126.

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

[16] C. Buckley, J. Allan, and G. Salton, "Automatic Routing and Ad-hoc Retrieval Using SMART: TREC-2", Proceedings of TREC-2 (1993), pp. 45-56.

[17] C. Buckley, G. Salton, J. Allan, and A. Singhal, "Automatic Query Expansion Using SMART: TREC-3", Proceedings of TREC-3 (1994), pp. 69-80.