TREC-4 Ad-Hoc, Routing Retrieval and Filtering Experiments using PIRCS

K.L. Kwok & L. Grunfeld

Computer Science Dept., Queens College, CUNY,
Flushing, NY 11367. email: kklqc@cunyvm.cuny.edu

ABSTRACT

Our ad-hoc submissions are pircs1 which is fully automatic, and pircs2 which involves manually weighting some terms and adding some new words to the original topic descriptions. The number of words added are minimal. Both methods involve training and query expansion using the best-ranked subdocuments from an initial retrieval as feedback. For our routing experiments we make use of massive query expansion of 350 terms in pircsL, with emphasis on expansion with low frequency terms. Training is done using short and top-ranked known relevant subdocuments. In pircsC, we define four different 'expert' queries (pircsL being one of them) for each topic by using different subsets of training document, and later combine their retrieval results into one. Filtering experiment is done with the retrieval lists of pircsL. For each query, we use the training collections to define retrieval status values (RSVs) where the utilities are maximum for the three precision types. These RSVs are then used as thresholds for the new collections. Evaluated results show that both ad-hoc and routing retrievals perform substantially better than median.

1. INTRODUCTION

The PIRCS retrieval system has been described in our previous TREC papers [KwPK93,KwGr94b,KwGL95] as well as in [Kwok95,90]. Basically, given a query q_a and documents d_i, we use two retrieval algorithms that give the following document-focused and query-focused ranking RSVs respectively:

$$RSV_{i/d} = \sum_k w_{ik} * w_{ka}$$
$$RSV_{i/q} = \sum_k w_{ak} * w_{ki}.$$

w_{ka} is the weight of term t_k used in q_a and is content oriented, and w_{ik} is the weight assigned to t_k based on its collection properties and is more discrimination oriented. (Similar for w_{ki} and w_{ak}). They are implemented as weighted edges connecting documents and queries to terms in a bi-directional network. The sum is over all terms common to d_i and q_a. Results of these two algorithms are then combined to return a final ranking RSV for document d_i as follows:

$$RSV_i = a*RSV_{i/d} + (1-a)*RSV_{i/q}$$

where a<1 is an adjustable parameter usually set to between 0.7 and 0.9. By viewing a document as not monolithic but composed of conceptual components each representable as a term and hence removing the 'binary' assumption of the normal probabilistic model, w_{ka} is given by $S(TermFreq_{ak}/Length_a)$, and correspondingly for w_{ki}. S(.) is a signal function to suppress outlying values and $Length_a$ is the length of q_a including repeating terms [Kwok90,95]. w_{ik} (and correspondingly for w_{ak}) consists of a trainable factor and an Inverse Collection Term Frequency factor ICTF = $\log [(TotalTerms-ColFreq_k)/ColFreq_k]$. ICTF is like the Inverse Document Frequency except that it also accounts for term frequencies, not just 'binary'. The trainable factor at the initial stage when queries and documents perform hard self-learning would be given by $\log [TermFreq_{ik}/(Length_i-TermFreq_{ik})]$. In our system we perform soft self-learning and w_{ik} has value lying between ICTF and ICTF + the hard self-learning value. When more documents known relevant (or assumed relevant) to q_a are available, this trainable factor can attain more accurate value and provide better retrieval results. Moreover, based on the training documents, edges connecting q_a to new terms may be added resulting in an expanded query. This enables retrieval of documents that do not share common terms with the original query. These processes of training, query expansion and retrieval are performed by sophisticated algorithms and are implemented as activation spreading in the network.

Sections 2 and 3 contain descriptions of our ad-hoc and routing experiments and in Section 4 the filtering experiment. Our conclusion is in Section 5.

2. AD-HOC RETRIEVAL EXPERIMENTS

2.1 Initial Queries

Ad-hoc experiments for TREC-4 differ from previous TRECs in that the new topic set Q5 (#202-#250) are deliberately short, consisting of one to three lines of text in the Description Section only. After automatic processing in our system with stopword removal and stemming, the number of unique terms left totaled 305, giving an average of only 305/49 = 6.22 unique stems per query. A distribution table of query sizes is shown in Fig.1 under pircs1. These small **initial** query sizes are not suitable for statistical systems like PIRCS which relies on accumulating evidence from many terms and phrases to define a topic of need. This has been shown to be true in [LuKe95]. Another related problem arising from these small query sizes is that the short descriptions generally do not offer opportunity for the query composers to repeat words, so that we also end up with queries having uniform, equally weighted terms with no differentiation of importance for representation. Since we will use query expansion for our **final** retrieval results (Section 3.2), the quality of the **initial** retrieval is important. In view of this, we decided to do three experiments: pircs1, 1a and 2, differing in the initial queries with which we start.

Pircs1 relies on initial queries obtained from the raw topics automatically via stemming and stopword removal. This gives the basis result from which we can try to improve. For pircs1a (unsubmitted), we manually weight the terms in the queries obtained in pircs1 by replicating some of the words. They are chosen based on our belief that these words represent concepts that are central to the topical needs. We are also careful to treat every replicated word separately so that we do not accidentally add 2-word adjacent phrases. In pircs2, we replicate words without the previous precaution, and also augment some queries with new words. They are manually added based on synonyms, acronyms, general-specific, or highly associated words that come to mind after reading the topics. No retrieval was involved. However, in keeping with the spirit of testing short queries, we only add minimally. After processing, a total of 52 terms are added to 30 queries, averaging 52/30 = 1.73 per (changed) query. 19 queries are left with the same words as in pircs1a. Generally we add to the shorter queries and at most three content words per query. These manual operations are different from those done in [BCCN95]: they expand the queries automatically first and then manually delete, re-group, re-weight some of the terms or add new terms from the narrative section; we re-weight and add terms prior to processing with no term deletion, and the query is later expanded automatically. The distribution of query sizes for pircs2 is also shown in Fig.1 with an average of 7.29 terms per query.

Number of Uniq.Terms	**Number of Queries pircs1**	**pircs2**
2	2	0
3	5	2
4	9	3
5	8	7
6	9	11
7	2	7
8	1	3
9	6	7
10	3	4
11	0	1
12	2	2
13	2	1
14	0	1
Average	6.22	7.29

Fig.1: Distribution of Query Sizes for Pircs1 and Pircs2

2.2 Disks 2 and 3 Textbase and Expanded Queries

Ad-hoc retrieval was done using the Tipster Disks 2&3 collections. They are separated into four subcollections with documents segmented into about 550-word chunks as in previous TRECs. They are served by a master lexicon of 643,755 unique terms that includes 55,599 entries of our semi-automatic 2-word phrases.

The queries discussed in Section 2.1 are used to do an **initial** retrieval on the textbase. The 40 best-ranked subdocuments of each query are then employed as 'relevant' documents to train and expand the initial queries automatically as in a routing situation. In

TREC-3, we used only 6 best-ranked items which was insufficient [RWJH95]. These 40 subdocuments of a query define a set of terms; the best 50 such terms are selected based on occurrence frequency (>=5) and the average probability of occurrence within the 40 documents. They are added to the initial query, resulting in an expanded query with average size of 52.4 terms. A second round of retrieval is done using the expanded queries, and the document ranking then constitutes our final results. This procedure is repeated separately starting with each of the three initial query types discussed in Section 2.1.

2.3 Results and Discussion

Results of our six experiments are summarized in Fig.2, where the * entries denote our official results. Percentage increases from the base values (%/%) are also shown horizontally (before and after query training and expansion) and vertically (between initial query types).

Initial Qry Type	**Initial Retrieval**	**After Expansion Retrieval**
pircs1 (auto)	**3327/.2015** (%/%) (% / %)	***3896/.2599** (17.1 / 29.0) (% / %)
pircs1a (manual)	**3642/.2259** (%/%) (9.5 / 12.1)	**4258/.2828** (16.9 / 25.2) (9.3 / 8.8)
pircs2 (manual)	**4044/.2619** (%/%) (21.6 / 30.0)	***4562/.3064** (12.8 / 17.0) (17.1 / 17.9)

Fig.2: Relev.Retr. @1000 / Non-Interpolated Precision Results Before and After Query Expansion for the 3 Initial Query Types Averaged over 49 queries.

Initial Retrieval Using Different Initial Query Types

It can be seen from the Initial Retrieval column of Fig.2 that by simply replicating some content words in the original topic statements improvements of 9.5% and 12.1% are obtained for relevants retrieved @1000 and the average non-interpolated precision (pircs1a 3642/.2259 vs pircs1 3327/.2015). Out of 49 queries, 38 led to equal or better results after weighting and 11 gave worse, a ratio of nearly 4:1. This operation is very simple and the choice of words are usually quite obvious with little intellectual effort because the topic descriptions are so short. With a good text editor this can be done in less than 1/2 minute per query. The recommendation is that users who submit short descriptions should be advised to replicate and append some content words, and this can buy a gain of roughly 10% from PIRCS.

When we also augment some of the topics with new words, larger gains of 21.6% and 30% in relevants retrieved @1000 and precision are observed (pircs2 4044/.2619 vs pircs1 3327/.2015). Out of 49 queries in pircs2, 41 lead to results equal or better than pircs1 and 8 worse, a ratio of 5:1. Thus, even though we add minimally, the effect on the initial retrieval is substantial. Finding words to augment the queries however takes more time than just selecting words to replicate, probably about 3 hours for 50 queries. Quite often, trying to think of new words to add and later abandoning the effort takes longer time than having obvious associated terms.

Retrieval Using Different Expanded Queries

From Fig.2 we observe that retrieval via our query expansion procedure using best-ranked subdocuments for training is quite successful, leading to improvements of 12.8% to 29.0% from the initial query results. The procedure for training and query expansion is fully automatic. Comparing horizontally, we see that when the basis is higher the improvement is less (e.g. pircs2 initial 4044/.2619 vs expansion 4562/.3064: 12.8%/17.0%; pircs1 initial 3327/.2015 vs expansion 3896/.2599: 17.1%/29.0%), which is not unexpected.

Comparing vertically, we see that if one starts with better initial queries that lead to better initial retrieval, the final query expansion retrieval is also better (pircs2 4562/.3064: 17.1%/17.9% better than pircs1 3896/.2599). Simply weighting the query content terms only produces about half the improvements of pircs2 from pircs1 (pircs1a 4258/.2828: 9.3%/8.8% better than pircs1 3896/.2599). Pircs2 has 32 queries better and 17 worse than pircs1, and for pircs1a 30 better and 19 worse in these expanded query retrieval results.

As noted above, the exercise of weighting and adding new words to queries does not always lead to better results. Examples of successes are: the added synonyms car, auto and automobile in queries 219, 230 and 237; rubber (associated with tire) in 203; coal (specific-general to fossil fuel) in 243. Examples of failures are: death sentence (synonym to capital punishment) in 222; militia group (associated with paramilitary) in 231; iron (associated with steel) in 218; newborn deaths (associated with infant mortality) in 215.

Using pircs2 results, we observe that it recalls (4562/6501) 70.2% of "all" relevants at 1000 documents retrieved. At 10 documents retrieved, one can expect more than 5 of them are relevant, and at 30 more than 13.

Comparison with Other TREC-4 Sites

Comparison with the MEDIAN values of TREC-4 submissions are summarized below in Fig.3. It can be seen that results of pircs1 and especially pircs2 substantially outperform the median. As before, we also calculate MAXI-retrieval as a hypothetical system that returns the best performance for each query among all sites. This assumes that we have an intelligent agent who is able to choose the best retrieval system among this set of participants for each query, and would reflect the best we can do using our collective wisdom at this time. This MAXI-system will return an average precision, precision at 100 docs and at 1000 docs of 0.4638, 0.4743 and 0.1115 respectively. Thus, pircs2 achieves 66.1%, 69.9% and 83.5% and pircs1 achieves 56.0%, 60.4% and 71.30% respectively of these best values. PIRCS appears to achieve good results at the high recall region.

	pircs1			**pircs2**		
	>	=	<	>	=	<
av. prec:	30(3)	3	16	40(6)	0	9
rel_ret @ 100:	32(3)	4	13	39(6)	1	9
rel_ret @ 1000:	39(8)	2	8	44(14)	2	3

(figure in parenthesis is number of queries equaling the best values)

Fig.3: Comparison of Ad-Hoc Results with the Median from All Sites

3. ROUTING RETRIEVAL EXPERIMENTS

3.1 Routing Queries

We submitted two routing retrieval results. The first one is PircsL, where L means large query. Following Buckley et.al. [BuAS94] we found that massive term expansion of queries is beneficial for our model too.

The second submitted retrieval, PircsC , involves an experiment with combination of retrievals. Combining queries has received a lot of attention lately. Lee tried combining different normalizations available in the SMART system [Lee95], Kantor combined natural language, hard boolean and inferential retrieval, (decision level fusion) using several different schemes [Kant95], Fox and Shaw combined different soft-boolean and vector retrievals [FoSh94], and we also combined soft-boolean with PIRCS retrieval in our previous TREC ad-hoc experiments. In TREC-4, our approach is different, in that we try to create queries by limiting the training data to certain collections. For example one set of queries was trained by using only relevant information from the Ziff and FR collections. The rationale for this is that certain collections may have better, more focused coverage of some topics. The other interesting idea is an attempt to predict the performance of each method for each query and combine them accordingly.

PircsL (Large Query)

For the TREC4 routing experiments our starting point was the methods developed for TREC3 [KwGL95]. The learning and retrieval algorithms remained the same. Documents are broken up into 550 word segments, and if more than one subdocument is retrieved, their retrieval status value is combined by the formula RSV=1.0 * first + .1 * second + .05 * third. Only relevant documents are used for training. Judged non relevants are ignored. Queries include terms from original query and terms expanded from relevants. Term weights are based on relevant documents.

Experiments after TREC3 revealed that our queries have very good recall, but the precision could be improved. The term selection formula was changed to favor low frequency terms. The term selection formula in TREC2 was sum (Wi / log [max (2000,DOCFREQi)])/N,

where Wi is the weight of term i in the relevant document defined as (count of term i)/(count all terms), N is the number of relevant documents, DOCFREQi is the document frequency of term i, the sum is over all relevant documents for that query. Note that this takes into account the within document frequency. The denominator favors higher frequency terms, which represent more general concepts. This was reasonable for the relatively small query expansion. For TREC3 we substituted 20 for 2000, ranking the low frequency terms much higher for selection. The submitted PircsL queries were expanded by 350 terms.

In the routing environment, where a large number of relevant documents of different sizes and quality are available, selecting a good subset for training queries is important, rather than using all. A number of strategies are described in [KwGr94]. Two methods were found to produce good results: (a) selecting short documents and (b) selecting the top ranked subdocument from each document.

(a) Selecting short documents has the advantage, that it does not require ranking, and the documents will not contain many other topics. Also documents that are not ranked high by our system will be included. The disadvantage is, that not all queries have enough short documents to produce a good query.
(b) Selecting top ranked subdocuments will give a more broad based sample to train on, but it is biased toward the systems own retrievals, and does not take full advantage of the information available from other relevants. In practice it is slightly better than selecting short documents.

For the pircsL query, we use a hybrid selection method. We selected all short documents (those with 550 words or less) and added to it the top 50 retrieved relevant subdocuments.

An important issue is the selection of the test collection. Creating the query first and testing them on the data on disks 1, 2 and 3 would not be appropriate, since that would be a retrospective retrieval, and good results there may not give good results on an unknown collection. In fact there is a danger of overfitting the test data. On the basis of the information available to us, we assumed that the routing retrieval will be on documents similar to the Ziff and Federal Register collections. Therefore we decided to evaluate the results, based on retrievals on the Ziff and FR collections on disk 2, and train the queries on the other collections. After the methods were selected, we added the Ziff and Fr relevants from disk 2 to the training documents, to create the official query.

PircsC (Combined Queries)

Combining retrievals are known to be beneficial, provided that the retrievals are independent and are about the same power. PIRCS has the ability to combine different methods in the network. The reasoning is, that relevant documents will occur more non-randomly, in the retrieval lists than irrelevants, therefore combining them could improve their ranking.

A person who reads only the Wall Street Journal will probably create a different query than a person who reads PC Magazine, given the needs and concepts contained in a topic. The hypothesis is, that the large number of training documents available from publications of differing perspectives will allow for the creation of independent query formulations by different 'experts'.

For pircsC we combined 4 different query formulations:

Expert #1: (pircsL) Long query, expansion 350. This is the same as pircsL.
Expert #2: (short) Short query, term expansion is only 80. The training method for this query favors larger frequency words. The training documents are the same as for #1. This query is very similar to our query at TREC-3.
Expert #3: (ZFD) Query was trained only on Ziff and Federal Register and DOE documents. Query expansion is 300.
Expert #4 (WAS) Query was trained only on WSJ AP and SJM documents, Query expansion 300.

There are a number of ways to add retrievals. A simplistic way is just to add the retrieval status value after some normalization, so that the rsv of different queries are compatible. In the routing environment we can get a reasonable estimate as to each method's performance for each query, and use this information to enhance the result.

Experimenting with combining two retrievals only, we found, that when they perform similarly, it is beneficial to add them, otherwise take the result of the better one. A function which has this desired behavior for large N is (Xi)**N, where Xi is the Av11 for the method i and N is an integer.

For the pircsC submitted query, we made 2 adjustments. Since for very low Av11 this expression goes very quickly to zero, we added a .1 to all Av11 values. Also, since estimates were made based on retrieval performance in Ziff and FR on disk2, the performance of the Expert #3 was probably underestimated, since it is missing 50 percent of it's training documents (since it was created using only disk1 at this point), therefore we added an extra 0.03 to all the av11 for this 'expert'. The value of N was 4.

3.2 Results

In all the results below we use the short query as the baseline, since it is closest to our TREC3 query, so we can assess any improvements on it.

	Rel Ret	**% chg**	**Avg prec**	**% chg**
short	5555	0.00%	0.3713	0.00%
pircsL	5645	1.62%	0.3901	5.06%
zfd	5501	-0.97%	0.3672	-1.10%
was	5406	-2.68%	0.3582	-3.53%

	Rel Ret	**% chg**	**Avg prec**	**% chg**
comb1111	5658	1.85%	0.3900	5.04%
comb8421	5626	1.28%	0.3924	5.68%
comb1	5658	1.85%	0.3926	5.74%
pircsC	5635	1.44%	0.3909	5.28%
comb8	5612	1.03%	0.3891	4.79%
comb200	5571	0.29%	0.3843	3.50%

Out of the 4 experts, the large query (pircsL) performed best, as was expected. It improved by 5% over the short query, which uses basically the same method that was submitted to TREC3. Although it is difficult to make comparisons accross different query collections, this seems to indicate, that the TREC4 queries are somewhat more difficult, since there the score was .388. The other 2 experts performance was worse. It is interesting to note that the ZFD expert, which was trained on a similar type of collection as the target did not do better. Perhaps the reason is that there were not enough training documents.

It is interesting to notice, that out of the 50 queries PircsL had 20 bests and the others about 10 each. This bears out the hypothesis, that while the individual experts may not be very good overall, they can become very good at individual queries.

A number of retrievals using different combinations were tried. All combination were combined using the equation RSVj= SUM (ai * RSVi j) , where ai is a constant for method i and RSV is the return status value for method i , for document j and the sum is over the methods i.

The different combinations vary the value of a. Comb1111 is what was called a simplistic addition, a=1. Comb8421 is defined as: a= 8 to the best expert a=4 to the second a=2 to the third and a=1 to the fourth. The idea is the we want the best retrieval to dominate and the other retrievals to provide corrections.

The other methods use the formula a = (Xi)**N , where Xi is the Av11 for the method i and N is an integer. Note that if N=1 then a=Av11 and if N=200 then the best retrieval has a=1 and the others 0, except in the case where the Av11 is very close to each other.

The other 4 combinations vary the N in the above formula. N=1,4,8 and 200 were tried.

The differences may not be significant to draw any conclusions. From the first 2 combinations it seems that its better to give more weight to the better expert. From the last 4 combinations we can conclude the opposite.

We still believe, that it is possible to create multiple experts from large number of relevant documents, perhaps by using more sophisticated methods, such as clustering. Also prediction of expert performance can be improved. Note that the combination with N=200, which takes the best performing expert for each query did worse the PircsL. Had we been able to predict the best expert for each query and use that query the result would be .4027, a 3% improvement.

Comparison with Other TREC-4 Sites

	pircsL			**pircsC**		
	>	=	<	>	=	<
av. prec:	41(3)	2	7	42(5)	0	8
rel_ret @ 100:	35(4)	8	7	44(6)	8	8
rel_ret @ 1000:	41(19)	7	2	41(19)	6	3

(figure in parenthesis is number of queries equaling the best values)

Fig.4: Comparison of Routing Results with the Median from All Sites

4. FILTERING EXPERIMENTS

We also participated in the filtering track. Our approach was to try to predict the rsv at which the filter evaluation function is the maximum. The training data used were the disk2 Ziff and FR document collections. We did another set using the entire disk2.

The retrieval was done using the pircsL query discussed above, therefore all training was done using retrospective retrieval. We think this has the effect of concentrating the relevants at the top of the retrieval, making the maximum filter evaluation point higher that what it should be, leading to a more conservative estimate, if there are a lot of relevent documents, the opposite may happen if there are only a few.

Training on the the entire disk 2, returned resulted at a much lower cutoff and more documents returned, but for the submission we decided to be conservative and submitted the result based on the Ziff and FR collections only.

At this point we do not have results on which method performed better.

5. CONCLUSION

The PIRCS retrieval system and its learning capabilities have consistently been demonstrated to give exemplary performance in ad-hoc and routing retrievals. In ad-hoc, in the absence of known relevants, we have shown that training and query expansion from a set of best ranked subdocuments from an initial retrieval is beneficial. The better the initial retrieval, the better the final results. Manually weighting terms in a given topic, and minimally adding a few new terms can lead to this better initial environment. In the future, we will try to automate these manual processes. For routing, we have shown that PIRCS can benefit from massive query expansion of over 300 terms. We also introduce a method of defining several ’expert’ queries from a single topic based on subsets of the relevant documents. Combining the retrieval results from several ’expert’ queries lead to slightly better results. More investigation needs to be done to find the best method for this operation.

ACKNOWLEDGMENTS

This work is partially supported by a PSC-CUNY grant.

REFERENCES

[BCCN95] Broglio, J, Callan, J.P, Croft, W.B & Nachbar, D.W. Document retrieval and routing using the INQUERY system. In: Overview of the Third Text REtrieval Conference (TREC-3). Harman, D.K. (Ed.). NIST Special Publication 500-225, 1995, pp. 29-38.

[BuAS94] Buckley, C., Allan, J. & Salton, G. Automatic Routing and Ad Hoc Retrieval using SMART: TREC2. In: The Second Text REtrieval Conference (TREC-2). Harman, D.K. (Ed.). NIST Special Publication 500-215, 1994, pp. 45-55.

[FoSh94] Fox, E. A and Shaw, J. A. Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2). Harman, D.K. (Ed.). NIST Special Publication 500-215, 1994, pp.243-252.

[Kant95] Kantor P.B Data Fusion in Information Retrieval. In: Overview of the Third Text REtrieval

Conference (TREC-3). Harman, D.K. (Ed.). NIST Special Publication 500-225, 1995, pp.319-332.

[KwGL95] Kwok, K.L, Grunfeld, L & Lewis, D.D. TREC-3 ad-hoc, routing retrieval and thresholding experiments using PIRCS. In: Overview of the Third Text REtrieval Conference (TREC-3). Harman, D.K. (Ed.). NIST Special Publication 500-225, 1995, pp.247-255.

[KwGr94a] Kwok, K.L. & Grunfeld, L. Learning from relevant documents in large scale routing retrieval. In: Proc. Human Language Technology Workshop, ARPA, Mar 8-11, 1994 Plainsboro, NJ. Morgan Kaufmann, San Francisco, 1994, pp. 358-363.

[KwGr94b] Kwok, K.L. & Grunfeld, L. TREC-2 Document retrieval experiments using PIRCS. In: The Second Text REtrieval Conference (TREC-2). Harman, D.K. (Ed.). NIST Special Publication 500-215, 1994, pp. 233-242.

[Kwok90] Kwok, K.L (1990). Experiments with a component theory of probabilistic information retrieval based on single terms as document components. ACM Transactions on Office Information Systems, 8:363-386.

[Kwok95] Kwok, K.L (1995). A network approach to probabilistic information retrieval. ACM Transactions on Office Information Systems, 13:325-353.

[KwPK93] Kwok, K.L., Papadopolous, L & Kwan, Y.Y. Retrieval experiments with a large collection using PIRCS. In: The First Text REtrieval Conference (TREC-1). Harman, D.K. (Ed.). NIST Special Publication 500-207, 1993, pp. 153-172.

[Lee95] Lee, J.H. Combining multiple evidence from different properties of weighting schemes. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp.180-188, 1995.

[LuKe95] Lu, X.A & Keefer, R.B. Query expansion/reduction and its impact on retrieval effectiveness. In: Overview of the Third Text REtrieval Conference (TREC-3). Harman, D.K. (Ed.). NIST Special Publication 500-225, 1995, pp.231-239.

[RWJH95] Robertson, S.E, Walker, S, Jones, S, Hancock-Beaulier, M.M, Gatford, M. Okapi at TREC-3. In: Overview of the Third Text REtrieval Conference (TREC-3). Harman, D.K. (Ed.). NIST Special Publication 500-225, 1995, pp.109-126.