MATRIX at the TREC 2005 Robust Track

W.S. Wong[1], H. C. Wu[1], R.W.P. Luk[1], K.F. Wong[2], K.L. Kwok[3], H.V. Leong[1]

[1]Department of Computing

The Hong Kong Polytechnic University

[2]Department of System Engineering and Engineering Management

The Chinese University of Hong Kong

[3]Department of Computer Science

City University of New York, USA

ABSTRACT

In the TREC 2005 robust retrieval track, we tested our adaptive retrieval model that automatically switches between the 2-Poisson model/adaptive vector space model and our initial predictive probabilistic context-based model depending on some query characteristics. Our 2-Poisson model uses the BM11 term weighting scheme with passage retrieval and pseudo-relevance feedback. The context-based model incorporates the term locations in a document for calculating the term weights. By doing this, different term weights are assigned to the same query term depending on its context and location in the document. We also use WordNet in the term selection process when doing pseudo-relevance feedback. The performance of our model is comparable to the median among all participants in the robust track on the whole query set including the title, descriptive and long queries.

1. INTRODUCTION

In the formal runs of the TREC 2005 robust retrieval track, we used the 2-Poisson model [1] or the (adaptive) vector space model, together with the probabilistic context-based retrieval model [2]. In the informal runs, we tested the performance of the 2-Poisson model and estimated the optimal performance of our model. Our search engine has migrated from UNIX to Linux and our search engine is now called MATRIX.

In natural language processing, there are problems due to polysemy and synonyms. Polysemy is a term with multiple meanings while synonyms are different terms with the same meaning. In the context of information retrieval, polysemy causes the degradation of precision since a query term found in a document may not carry the same meaning as in the query. That is, the spelling of a term matches but its meaning does not match. The similarity score of the document with the query term is erroneously increased. The problem of synonyms is that the term used by the author of the document may be different from that used by the user of the information retrieval system while they both refer to the same meaning. That is, the spelling of the terms does not match but their meaning matches. This will cause a decrease in recall as the similarity score of the document is erroneously decreased. These problems can be generalized to the problem of finding term dependencies. Work has been done to solve the problem by using WordNet [3-5] or/and co-occurrence of query terms in a document [6-9]. For the problem of synonyms, we use WordNet to find terms with similar meaning like the previous work. In contrast to the previous work, we solve the problem of polysemy by considering the location of the query terms in a document when calculating the term weights; this is our context-based model.

The rest of the paper is organized as follows. Section 2 describes the models used in our formal runs in the robust retrieval track and the performance of the runs. Section 3 presents the performance of the first part of our informal runs which use the 2-Poisson model solely. Section 4 is the second part of our informal runs. We tested our model retrospectively (i.e., when relevance judgment is present) in order to estimate the optimal performance of our vector space model. Section 5 provides a conclusion.

2. FORMAL RUNS

In our formal runs, we adaptively switch between two models, namely the passage-based 2-Poisson model with BM11 term weighting scheme [1] or the adaptive vector space model (AVSM) combined with the predictive version of the probabilistic context-based retrieval model [2].

2.1. Passage-Based 2-Poisson Model

In our 2-Poisson model, we use the BM11 term weighting scheme with passage-based retrieval. Pseudo-relevance feedback (PRF) is also performed for expanding the query after the first pass retrieval. Each passage has a fixed length of 300 terms, unless the end of file is encountered. The document similarity score *sim(.)* is computed by combining passage scores using a weighted Boolean disjunction operation [10] or generalized mean function, conforming to the DRD principle [11]:

$$sim(d_i, q) = \sqrt[a]{\frac{1}{k_i}\sum_{j=1}^{k_i} rel(p_{i,j}, q)^a}$$

where q is the query, d_i is the i-th document, $p_{i,j}$ is the j-th passage of the i-th document, k_i is the number of passages in the i-th document, *rel(.)* is the relevance score assigned by the 2-Poisson retrieval model with BM11 term weights, and a (=20) is a soft-hard decision parameter.

From the experiment results of the past TREC data collections, using pseudo-relevance feedback can improve the retrieval performance. However, the parameters (e.g., number of feedback terms) in pseudo-relevance feedback should be carefully set in order to have performance gain. In the TREC 2005 robust track, we use the top N (=20) documents from the first pass retrieval for selecting the feedback terms. Forty top ranked terms in the retrieved documents are selected for expanding the query, and then a second pass retrieval is performed using the expanded query.

2.2. Probabilistic Context-Based Model

In order to solve the problem of polysemy, we consider the context of a query term for weighting the query term in a document. We would like to differentiate the meaning of context here with the meaning of the user context analysis [12]. We believe that the meaning of a term is highly related to its context terms, that is, for a term which has two different meanings, say meaning *A* and meaning *B*, the occurrence of the context terms for meaning *A* should be quite different from the occurrence of the context terms for meaning *B*. Intuitively, the meaning of a term can be determined by looking into where the term is used, that is, the context of the term.

Define $t_{i,k}$ to be the term occurred at the k-th location of the i-th document. If $t_{i,k}$ is a query term, then we denote it $q_{i,k}$, a query term occurred at the k-th location of the i-th document, where $q_{i,k}$ is equal to $t_{i,k}$. For a query term $q_{i,k}$, a context $c(q_{i,k}, n)$ is defined as a window of terms with size n (i.e., n-term window) which the slots of the window follows the requirement below (in our robust track experiments, n is set to 31):

$$c[j] = \begin{cases} t_{i,k-(0.5n-j)} & \text{if } j < ceiling(0.5n) \\ q_{i,k} & \text{if } j = ceiling(0.5n) \\ t_{i,k+(j-0.5n)} & \text{if } j > ceiling(0.5n) \end{cases}$$

where $j \in [1, n]$ and the function *ceiling(.)* takes a real number x and returns the smallest integer that is greater than or equal to x. Strictly speaking, the context of a query term $q_{i,k}$ occurred at the k-th location of the i-th document is the terms surrounding and including $q_{i,k}$.

Using the notion of the context, we can develop a probabilistic context-based retrieval model [2]. We calculate the log-odds ratio of the probabilities of relevant and irrelevant given a particular context and assign the value to the query term weight. This is similar to the famous probabilistic model proposed by Sparck Jones et al [13].

$$w(q_{i,k}) = \log\left(\frac{P(relevant \mid c(q_{i,k}, n))}{P(irrelevant \mid c(q_{i,k}, n))}\right)$$

Using Bayse' rule,

$$\frac{P(relevant \mid c(q_{i,k},n))}{P(irrelevant \mid c(q_{i,k},n))} = \frac{P(c(q_{i,k},n) \mid relevant)}{P(c(q_{i,k},n) \mid irrelevant)} \times \frac{P(relevant)}{P(irrelevant)}$$

Since $P(relevant)$ and $P(irrelevant)$ are constants, their ratio is also a constant and can be ignored for the purpose of ranking. The term weighting function becomes:

$$w(q_{i,k}) = \log\left(\frac{P(c(q_{i,k}, n) \mid relevant)}{P(c(q_{i,k}, n) \mid irrelevant)}\right)$$

Like many other probabilistic models, we assume that the terms inside a context are independent to each other, so that we can multiply the probabilities of individual context terms. The probabilities of seeing a context term given the relevant and irrelevant term sets are calculated by the relative frequencies estimates of that term inside relevant and irrelevant term sets respectively. Since each document may contain more than one context, we need to aggregate the term weights of the contexts in order to determine the score of the document. There are various ways of aggregating the query term weights, such as averaging them, adding them together, picking the maximum and picking the minimum. We use the maximum weight as the score of the document that is consistent with the DRD principle [11]:

$$sim(d_i, q) = \max_{q_{i,k} \in q} \{w(q_{i,k})\}$$

where q is the query, d_i is the i-th document and $q_{i,k}$ is the query term occurred at the k-th location of the i-th document.

The retrospective experiments (i.e., relevance information is present) in [2] showed that context information can improve the retrieval performance. However, as we do not have the relevance judgments of the TREC 2005 robust track in our formal runs, we need to estimate the probabilities of relevant and irrelevant of a particular context or a particular term. Originally in the retrospective experiments, we use the relative frequency estimates to estimate the probabilities $P(t_{i,k} \mid relevant)$ and $P(t_{i,k} \mid irrelevant)$. In the predictive experiments, we should either estimate the probabilities directly (e.g., using relevance-based language model [14]) or estimate the sets of relevant and irrelevant terms. We adopt the latter approach in our robust track experiments. In order to estimate the relevant term set, we use the top N (=10) documents from the first pass retrieval, then we extract the contexts in these documents and the context terms are our estimated relevant terms. Similarly for the irrelevant term set, we use the bottom M (=100) documents for doing the estimation. We use a smaller number of documents for estimating the relevant term set than the irrelevant term set because we need an accurate relevant term set with as little noise as possible in order to have good results.

2.3. Adaptive Switching Model

The problem of our context-based model is that when the number of contexts in the top N retrieved documents is small, the size of the estimated relevant term set decreases. This will cause the problem of data scarcity as in the language modeling approach, as many of the terms are unseen terms, they will be assigned zero probabilities which is not desirable. Smoothing is one approach to tackle the problem [15]. Another approach is to use an adaptive model to switch between the 2-Poisson model and the context-based model, if the number of contexts found in the top N retrieved documents is small, we do not use the context-based model but the 2-Poisson model for ranking the documents. This forms our basic model in the formal runs of the robust track.

2.4. Performance of Our Formal Runs

Table 1 shows the performance of our formal runs in the TREC 2005 robust track while Table 2 compares our performance with all the participants' performance in the robust track. Our performance is slightly better than the median for the title and long queries while the performance of the descriptive queries is comparable to the median of all participants. The runs with an infix "2" are the runs using 2-Poisson model and the runs with the infix "V" are the runs using the AVSM. The HKPUCD run uses only the context-based retrieval model.

Figure 1 and 2 shows the difference in performance for each query between our formal runs' MAP and all participants' median MAP in title and long queries respectively. The difference for a query is simply calculated by subtracting the median MAP of all participants from our MAP for that query. From the results, we can discover that our performance is worse than the median MAP of all participants for a particular set of queries such as query 325 (Cult Lifestyles), 426 (law enforcement, dogs) and 427 (UV damage, eyes). Queries 426 and 427 have a common characteristic that they are combination of two different concepts. Further investigation is needed for the reasons of the decrease in performance in these queries.

Table 1: Performance of our five formal runs

Run Name	*MAP*	*P@10*	*P@30*	*R-Precision*	*GMAP*
HKPUVCT	0.248	0.442	0.422	0.293	0.133
HKPU2CT	0.246	0.426	0.411	0.291	0.129
HKPUCD	0.176	0.386	0.350	0.235	0.105
HKPUVCTDN	0.252	0.448	0.426	0.300	0.139
HKPU2TDN	0.244	0.422	0.403	0.289	0.123

Table 2: Comparison of our formal runs with all participants' runs

Run Name	*Query Type*	*MAP*				*P@10*			
		Our Runs	*All Participants' Runs*			*Our Runs*	*All Participants' Runs*		
			Best	*Median*	*Worst*		*Best*	*Median*	*Worst*
HKPUVCT	T	.248	.332	.223	.000	.442	.592	.434	.010
HKPU2CT	T	.246				.426			
HKPUCD	D	.176	.289	.183	.028	.386	.536	.386	.096
HKPUVCTDN	TDN	.252	.332	.218	.000	.448	.628	.432	.010
HKPU2TDN	TDN	.244				.422			

Figure 1: Difference in performance between HKPUVCT and all participants' median MAP in title queries

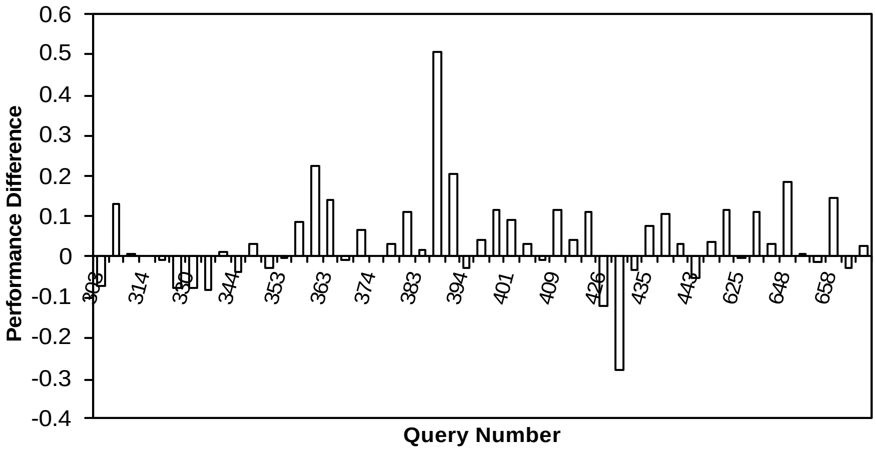
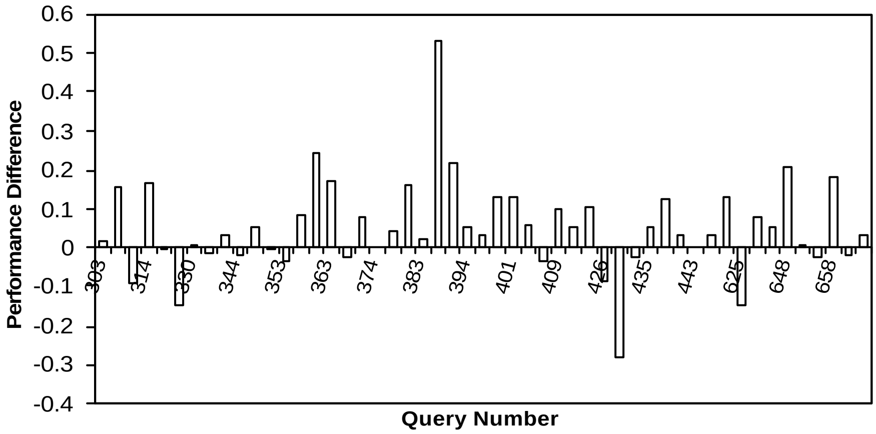


Figure 2: Difference in performance between HKPUVCTDN and all participants' median MAP in long queries



2.5. Storage Cost

The TREC 2005 data collection consists of 1,033,461 documents and consumes about 5 Gigabytes (block size). There are 1,212,595 passages where each passage has at most 300 terms. We use Elias delta code and Elias gamma code for the compression of the documents and term frequencies respectively [16]. Table 3 shows the storage size of the dictionary, document information and our extensible inverted index with compression. The indexing time was about 2 hours. This is achieved with an un-optimized index program code (e.g., includes many software flags, double document scanning, etc.).

Table 3: Size of dictionary, document information and extensible inverted index with compression

	Dictionary	*Document Information*	*Extensible Compressed Inverted Index*	*Total*
Size (MB)	97	160	332	589

3. INFORMAL RUNS

In the first part of our informal runs, we solely tested our passage-based 2-Poisson model on the TREC 2005 data collection. We also use WordNet to find the related terms for query expansion in the pseudo-relevance feedback. The results of the runs are showed in Table 4 (**HKPUINF1** and **HKPUINF2**). It is noticed that the performance of our passage-based 2-Poisson model is better than the adaptive switching model in our formal runs. The performance difference between our informal runs' MAP and all participants' median MAP is statistically significant for the long queries (p=0.01), however, the title queries do not have statistical significant improvement (p>0.1). The result reveals that using our context-based model has a negative performance gain actually. The main reason for this may be due to the incorrect estimation of the relevant and irrelevant term sets in our context-based model. Also, the assumption that the terms inside a context are independent to each other is obviously unrealistic.

In **HKPUINF3**, we merge the retrieval list of **HKPUINF2** (*L1*) with the retrieval list of the fuzzy Boolean model with passage-based retrieval on the conjunction of title queries (*L2*). The number of documents in *L2* is less than that in *L1* because it requires all title query terms to appear in a particular passage in order to be retrieved. The scores of the documents in the retrieval lists are normalized to be between 0 and 1. For a particular query, define $S_{i,L1}$ to be the score of document *i* in *L1*. If document *i* does not in *L1*, then $S_{i,L1} = 0$, similarly for $S_{i,L2}$. Further define $r = |L1 \cap L2| / |L2|$ which is the ratio between the number of documents found in both lists and the number of documents in $L2$. If $r < d, d \in [0,1]$, we perform a linear interpolation between $S_{i,L1}$ and $S_{i,L2}$ using the weight l :

$$S_i = l \times S_{i,L1} + (1 - l) \times S_{i,L2}$$

After the experiments, we found that $d = 0.4$ and $l = 0.9$ produce the best result with a slight improvement over **HKPUINF2**.

Table 4: Performance of our informal runs

Run Name	*Query Type*	*MAP*	*P@10*	*P@30*	*R-Precision*	*GMAP*
HKPUINF1	T	0.250	0.432	0.410	0.288	0.133
HKPUINF2	TDN	0.301	0.554	0.488	0.334	0.216
HKPUINF3	TDN	0.306	0.574	0.494	0.341	0.225

4. RETROSPECTIVE INFORMAL RUNS

Our experiments in these informal runs focused on estimating the retrieval effectiveness based on a retrospective study that makes use of the formulae in Relevance Feedback (RF).

4.1. SETUP

In our experiments, we used the vector space model (VSM) as our retrieval model because the VSM and RF are based on the same idea that queries and documents are modeled as vectors in the hyperspace of term weights. Therefore, VSM is consistent with RF conceptually. Our VSM uses the pivoted unique normalization [17] to compute the similarity score between the query and the document. Our similarity calculation was similar to the one used by AT&T in TREC [18] except the query weight was calculated by query term frequency rather than *ltu* formula. Our VSM model in these runs are not based on passages but based on the whole documents. The index terms are found in the documents as strings between two (white) space characters. Unwanted words were filtered using a list of 441 stop words and candidate index terms are stemmed by the Porter stemming algorithm [19].

Table 5 shows the retrieval effectiveness of our fourth and fifth informal runs. The fourth run is labeled **HKPUINF4** and the fifth run is labeled **HKPUINF5**. Title queries were used for these two runs and the pseudo-relevance feedback (PRF) was applied in the **HKPUINF5** run but not in the **HKPUINF4**. In the PRF cycle, the new query was formulated by the title query and top 40 terms which was selected from top 10 documents in the retrieval list. The main purpose of these runs is to investigate the baseline performance of our system. The measures used for assessing the performance are MAP, P@10, P@30 and R-Precision. The MAP performance of the system is clearly lower than the formal runs using VSM with the adaptive pivoted document length normalization based on passage retrieval (Table 1) but the GMAP performances of the passage and document retrievals are similar.

Table 5: The baseline performance of our system using title queries

Run Name	*MAP*	*P@10*	*P@30*	*R-Precision*	*GMAP*
HKPUINF4	0.173	0.366	0.318	0.237	0.1035
HKPUINF5	0.220	0.450	0.380	0.262	0.1386

4.2. Performance Limit of using Relevance Feedback

Relevance Feedback (RF) is a popular and effective query reformulation technique for improving retrieval performance since its initial conception by Rocchio [20] in the 1960's. RF modifies the query iteratively, based on the user's judgments of the top retrieved documents. Many researchers have tried to improve the effectiveness of RF and many were interested to find a good term selection method for RF. However, the estimation of the best retrieval effectiveness of RF itself is still unknown. Since TREC have published all the relevant documents for each topic, these results can help us to estimate the best retrieval effectiveness of RF more accurately. It can be thought of as the user who examines the entire retrieval list rather than just the top ten or twenty documents for a single RF iteration.

Briefly, our algorithm for estimating the performance limit of RF is similar to the PRF but takes all the relevant documents in a single iteration rather than taking top *N* documents in many RF iterations. Besides, the stop words, numerals and the terms with the occurrence in collection less than two are filtered. This is designed to avoid formulating trivial optimal queries where each relevant document can be potentially picked up by one term that only occurred in that relevant document. Our term ranking function is based on the common term weight function W4 [21]. Furthermore, the top 100 terms in the term ranking list and the 'title' part in the topic are combined to formulate the new query for each topic. The reason for using W4 and choosing the same query length for all topics is because we want to investigate the average performance limit of using RF rather than analyzing the performance limit of each topic.

4.3. Experiments in Robust Track

The experimental results using our algorithm on the dataset of robust track this year is shown on Table 6 and is labeled **HKPUINF6**. From the table we can see that, the MAP and P@10 of this run is 0.546 and 0.936 respectively. It is far beyond our performance in Table 5 and the best performance on TREC automatic formal runs as well as manually assisted formal runs (i.e., 0.332 for MAP and 0.628 for P@10). It means that there is still room to formulate a better query or "near optimal" query for current existing retrieval system to improve its effectiveness.

Table 6: The estimated performance limit of using Relevance Feedback in Robust Track

Run name	*MAP*	*P@10*	*P@30*	*R-Precision*	*GMAP*
HKPUINF6	0.546	0.936	0.815	0.543	0.526

5. CONCLUSION

In this year' s robust track, we tested our probabilistic context-based retrieval model with the passage-based 2-Poisson model or with the adaptive vector space model. In our formal runs, the performance is comparable to the median of all participants in the robust track. While in our first part of informal runs, the performance is better than the median of the performance of all participants in the robust track and the difference is statistically significant for the long queries (p=0.01). The results indicates that further investigation is required in order to come up a more accurate estimation of relevance and irrelevance models for the context-based retrieval model. In our second part of informal runs, we tested the optimal performance of our model retrospectively and the result indicates that there is still room for current models to improve.

ACKNOWLEDGEMENT

We would like to thank the Center for Intelligent Information Retrieval, University of Massachusetts (UMASS), for facilitating Robert Luk to develop in part the basic IR system, when he was on leave at UMASS. This work is supported by the CERG Project # PolyU 5183/03E and the extensible inverted index compression is supported by the project # A-PE36.

REFERENCES

[1] S. E. Robertson, S. Walker, M. M. Beaulieu, M. Gatford and A. Payne. Okapi at TREC-4. In *Proceedings of Fourth Text REtrieval Conference (TREC-4)*, pp 73-96, 1996.

[2] H. C. Wu, R. W. P. Luk, K. F. Wong, K. L. Kwok and W. J. Li. A retrospective study of probabilistic context-based retrieval. In *Proceedings of ACM SIGIR 05*, pp. 663-664, 2005.

[3] G. Cao, J. Y. Nie and J. Bai. Integrating word relationship into language models. In *Proceedings of ACM SIGIR 05*, pp. 298-305, 2005.

[4] S. Liu, F. Liu, C. Yu and W. Meng. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In *Proceedings of ACM SIGIR 04*, pp. 266-272, 2004.

[5] R. Mandala, T. Takenobu, and T. Hozumi. The use of WordNet in information retrieval. In *COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systert*, pp. 31-37, 1998.

[6] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proceedings of ACM SIGIR 05*, pp. 472-479, 2005.

[7] J. Gao, J. Y. Nie, G. Wu and G. Cao. Dependency language model for information retrieval. In *Proceedings of ACM SIGIR 04*, pp. 170-177, 2004

[8] R. Nallapati and J. Alan. Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 383-390, 2002

[9] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of ACM SIGIR 99*, pp. 222-229, 1999.

[10] E. Fox, S. Betrabet, M. Koushik and W. Lee. Extended Boolean models. *Information Retrieval: Data Structures and Algorithms*, pp. 393-418, 1992.

[11] Y. K. Kwong, R.W.P. Luk, W. Lam, K.S. Ho and F.L. Chung, Passage-based retrieval based on parameterized fuzzy operators, *ACM SIGIR Workshop on Mathematical/Formal Methods for Information Retrieva*l, 2004.

[12] P. Ingwersen and N. Belkin. Information retrieval in context - IRiX: workshop at SIGIR 2004 – Sheffield. In *ACM SIGIR 04 Forum*, pp. 50-52, 2004.

[13] K. Sparck Jones, S. Walker and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, 36(6), 779-808, 2000.

[14] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of ACM SIGIR 01*, pp. 120-127, 2001.

[15] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2), 179-214, 2004.

[16] H. Williams and J. Zobel. Compressing integers for fast file access. *The Computer Journal*, 42(3), 193-201, 1999.

[17] Singhal, C. Buckley and M. Mitra. Pivoted document length normalization. In *Proceedings of ACM SIGIR 96*, pp. 21-29, 1996.

[18] Singhal, J.Choi, D. Hindle, D.D. Lewis and F. Pereira. AT&T at TREC-7. In *Proceedings of TREC-7*, pp. 239-252, 1999.

[19] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3), 130-137, 1980.

[20] J. J. Rocchio. Relevance feedback in information retrieval. In *G. Salton editor, The SMART Retrieval System-Experiments in Automatic Document Processing*. Prentice Hall Inc., 1971.

[21] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Sciences*, 27(3), 129-146, 1976.