

University of Minnesota

This is to certify that I have examined this copy of
master's thesis
by

Inderjit Singh

And have found that it is complete and satisfactory in all
respects, and that any and all revisions by the final
examining committee have been made.

Carolyn J.Crouch

Name of Faculty Advisor

Signature of Faculty Advisor

Date

Graduate School

**The Impact of Phrases
on the
Retrieval Effectiveness of Very Short Queries**

A thesis
submitted to the faculty of the graduate school
of the University of Minnesota
By

Inderjit Singh

In partial fulfillment of the requirements
for the degree of
Master of Science

August 2002

Department of Computer Science
University of Minnesota Duluth
Duluth, Minnesota 55812
U.S.A.

Abstract

Relevance feedback and pseudo-feedback are considered very powerful techniques to improve upon the baseline for very short query retrieval. Both techniques require reformulation of the query, called the feedback query. The feedback query is generated by adding terms to the query vector from the documents retrieved in the first retrieval run. Our research concentrates on improving this set of documents (i.e., those used for query reformulation), by using phrases both from the original query and the documents retrieved. A set of *good phrases* is chosen for each query and the documents retrieved from the first iteration are reranked using these phrases. The purpose of reranking is to retrieve more relevant documents in the top-ranks so that a better feedback query can be generated. For the pseudo-feedback experiments, our reranking improved upon the previous reranking schemes although the results for the feedback run were not as high as expected. We were also able to achieve an improvement over the regular relevance feedback but the increase was not consistent over the collections. Trec6, Trec7 and Trec8 were used for these experiments, and retrieval effectiveness was measured using P@20 and Avg P.

Acknowledgements

I am grateful to many people for their support in this work. In particular I did like to thank:

Dr.Carolyn Crouch, for her guidance throughout this work. I am grateful to her for the invaluable feedback on my written work and this wonderful opportunity to extend the work of Steven Holtz and Qyingen Chen.

Dr.Donald Crouch and Dr.Robert McFarland for their suggestions and feedback.

Steven Holtz for helping me understand the previous work and filling me in with the implementation-level details of Smart system.

Dr.Ted Pedersen and Satanjeev Banerjee for helping me in the use of NSP.

Table of Contents

1. Introduction.....	2
1.1 Background.....	2
1.2 Related Work.....	3
1.3 Test Collection and Retrieval System.....	4
1.4 Holtz's Work.....	6
2. Initial Experiments.....	8
2.1 Using Unstemmed Terms and Phrases.....	8
2.2 Parameters for the Feedback Run.....	9
2.3 Results.....	9
2.3.1 Reranking.....	10
2.3.2 Reranked Pseudo Feedback.....	12
2.3.3 Reranked Pseudo Feedback(γ).....	14
2.4 Conclusions.....	16
3. Phrase Selection.....	17
3.1 Motivation.....	17
3.1.1 Holtz's Method of Phrase Selection.....	17
3.1.2 Problem with Holtz's Method.....	17
3.2 The N-gram Statistics Package.....	18
3.3 Our Method of Phrase Selection.....	19
3.3.1 Phrases from the Query.....	19
3.3.2 Phrases from Top-ranked Documents.....	20
4. Experiments with Relevance Feedback.....	23
4.1 Relevance Feedback.....	23
4.2 Hybrid Feedback.....	25
4.2.1 Phrase Selection.....	27
4.2.2 Reranking.....	28
4.2.3 Results.....	29
4.2.4 Conclusions.....	31
4.3 Retrieval with Phrases in Query Vector.....	32
5. Reranking and Reranked Pseudo-Feedback with Assumed	

Relevance for Phrase Selection.....	34
5.1 Phrase Selection.....	34
5.2 Reranking.....	35
5.3 Results.....	35
5.3.1 Reranking.....	36
5.3.2 Reranked Pseudo-Feedback.....	38
6. Conclusion and Future Work Suggestions.....	40
6.1 Experiments with Relevance Feedback.....	40
6.2 Experiments with Pseudo-Feedback.....	40
6.3 Future Work.....	43
7. References.....	44
8. Appendix.....	45
8.1 Reranking of the Phrase List given by NSP.....	45
8.2 Results.....	49

List of Tables

Table 1: Query Length Statistics.....	5
Table 2: Number of Queries Per Collection Binned on Length after Stopping.....	5
Table 3: CL Reranking compared to the Baseline.....	11
Table 4: SP Reranking compared to the Baseline.....	11
Table 5: IC Reranking compared to the Baseline.....	12
Table 6: RRPf with CL Reranking compared against baseline; $\alpha=8, \beta=8, \gamma=0$	13
Table 7: RRPf with SP Reranking compared against baseline; without negative feedback, $\alpha=8, \beta=8, \gamma=0$	13
Table 8: RRPf with IC reranking compared against baseline; without negative feedback , $\alpha=8, \beta=8, \gamma=0$	14
Table 9: RRPf with CL reranking compared against baseline; with negative feedback , $\alpha=8, \beta=8, \gamma=8$	15
Table 10: RRPf with SP reranking compared against baseline; with negative feedback , $\alpha=8, \beta=8, \gamma=8$	15
Table 11: RRPf with IC reranking compared against baseline; without negative feedback , $\alpha=8, \beta=8, \gamma=8$	16
Table 12: Relevance Feedback with varying R, $\alpha=8, \beta=8$	24
Table 13a: Comparison of Relevance Feedback and Hybrid Feedback (with bad phrases); M=10, R=5, $\alpha=8, \beta=8$	30
Table 13b: Comparison of Relevance Feedback and Hybrid Feedback (without bad phrases); M=10, R=5, $\alpha=8, \beta=8$	31
Table 14: Avg P and P@20 for Base Retrievals with and without phrases included in the query vector.....	32
Table 15: Holtz's SP Reranking and our SP reranking compared to the baseline.....	37

Table 16: RRPf with SP reranking (Holtz's and NSP) compared against baseline; without negative feedback, $\alpha=8, \beta=8, \gamma=0$...	39
Table 17: Comparison of RF, HF and HFb (Standard parameters for feedback as in [4], $T=300, \alpha=8, \beta=8$).....	41
Table 18: Comparison of PF, RF, RR and RRPf.....	42
Table 19: Top 20 phrases before and after reranking for the query - 'international organized crime'.....	48
Table 20: Reranking with no feedback (Holtz's Method)....	49
Table 21: RRPf; no negative feedback; $\alpha=\beta=8, \gamma=0$ (Holtz's method).....	50
Table 22: RRPf; $\alpha=\beta=\gamma=8, T=300$; Assumed non-relevant set=rank 501-100.....	51
Table 23: Rel. Fdbk. and Hybrid Fdbk; $R = 5$ and $M = 10$	52
Table 24: Rel. Fdbk. and Hybrid Fdbk; $R = 10$ and $M = 20$	53
Table 25: Rel. Fdbk. and Hybrid Fdbk; $R = 5$ and $M = 10$ (No bad phrases).....	54
Table 26: Rel. Fdbk. and Hybrid Fdbk; $R = 10$ and $M = 20$ (No bad phrases).....	55
Table 27: RR; $P_2=5, P_3=0$ and RRPf; no negative feedback; $\alpha=\beta=8, \gamma=0$	56
Table 28: RR; $P_2=5, P_3=5$ and RRPf; no negative feedback; $\alpha=\beta=8, \gamma=0$	57
Table 29: RR; $P_2=10, P_3=5$ and RRPf; no negative feedback; $\alpha=\beta=8, \gamma=0$	58
Table 30: RR; RRPf; $\alpha=\beta=8, \gamma=8$	59
Table 31: Trec6; PF, RF, Hybrid Feedback; Queries helped and hurt based on P@20.....	60
Table 32: Trec6; PF, RF, RR and RRPf; Queries helped and hurt based on P@20.....	61
Table 33: Trec7; PF, RF, Hybrid Feedback; Queries helped	

and hurt based on P@20.....62

Table 34: Trec7; PF, RF, RR and RPPF; Queries helped and hurt based on P@20.....63

Table 35: Trec8; PF, RF, Hybrid Feedback; Queries helped and hurt based on P@20.....64

Table 36: Trec8; PF, RF, RR and RPPF; Queries helped and hurt based on P@20.....65

Abstract

Relevance feedback and pseudo-feedback are considered very powerful techniques to improve upon the baseline for very short query retrieval. Both techniques require reformulation of the query, called the feedback query. The feedback query is generated by adding terms to the query vector from the documents retrieved in the first retrieval run. Our research concentrates on improving this set of documents (i.e., those used for query reformulation), by using phrases both from the original query and the documents retrieved. A set of *good phrases* is chosen for each query and the documents retrieved from the first iteration are reranked using these phrases. The purpose of reranking is to retrieve more relevant documents in the top-ranks so that a better feedback query can be generated. For the pseudo-feedback experiments, our reranking improved upon the previous reranking schemes although the results for the feedback run were not as high as expected. We were also able to achieve an improvement over the regular relevance feedback but the increase was not consistent over the collections. Trec6, Trec7 and Trec8 were used for these experiments, and retrieval effectiveness was measured using P@20 and Avg P.

1.Introduction

1.1 Background

The objective of our research is to improve the retrieval effectiveness of very short queries. These very short queries are two to three words long (though few of them are one or four words long). We use two measures to evaluate retrieval effectiveness: P@20 and Avg P. P@20 takes into account only the top 20 documents retrieved whereas Average Precision is a measure of overall performance.

Here we extended the work of Holtz [6] and Chen [3]. This work has been published in the paper entitled "Improving the Retrieval Effectiveness of Very Short Queries" by C.Crouch, D.Crouch, Q.Chen and S.Holtz. Holtz and Chen both worked on improving pseudo-feedback. A pseudo feedback run consists of two retrieval runs, where the terms from the top-ranked documents of the first retrieval are used to expand the query and the second retrieval is performed using the expanded query. Their approach to improving overall retrieval performance was to improve the pseudo-relevant set by reranking the documents obtained after the first retrieval run. Since their method involved reranking of documents before the feedback run, the method is called reranked pseudo-feedback (RRPF). Chen's reranking methods were completely based on single query terms, whereas Holtz's reranking methods were based on both single query terms and syntactic phrases.

In Holtz's work, reranked pseudo-feedback showed a remarkable boost in both P@20 and Avg P in comparison to base pseudo-feedback. Our work started with investigating the difference in RRPF by using unstemmed query terms for reranking instead of the stemmed terms. We continued with investigating different methods of phrase selection to be used for reranking. In the earlier work of Holtz [6], the phrase-based reranking methods used phrases obtained from the very short queries only. After stop-wording, very short queries are only two-three words long so there aren't many phrases to start with. We explored some new ways of finding good phrases for each query from the documents. We used a tool named the N-gram Statistical Package, developed by Pederson and Banerjee [10], for phrase selection. We also tried to improve on relevance feedback by using various methods based on *good phrases* obtained by the N-gram Statistical Package.

1.2 Related Work

Phrases have been investigated by many researchers for improving retrieval effectiveness, although their approaches vary in the phrase selection algorithm. The approach of Mitra *et. al.* [9] is to use a static list of phrases for the entire collection. This is very different from our method of selecting phrases on a query by query basis dynamically. Gey and Chen [5] and Mitra *et. al.* [9] used grammar rules to identify phrases. Our method of phrase selection is based purely on the frequencies with which two

words co-occur in the documents and queries. Our method is somewhat similar to [5], but they used Mutual Information as a measure to identify phrases in contrast to our Left Fisher's Test of Association. [9] also analyzed the difference between grammar and frequency-based phrases; for these experiments both methods of phrase selection yielded comparable results.

In our research, we use phrases for reranking purposes, unlike Kraaij and Pholmann [7] and Zhai [14] where documents were indexed with phrases. Most earlier research efforts centered on Trec1-Trec6, but our research involved extensive use of Trec6, Trec7 and Trec8.

1.3 Test Collection and Retrieval System

We used Text Retrieval Conference (TREC) document and query collections for all of our experiments (i.e., TREC-6, TREC-7 and TREC-8). Each of these collections comes with a set of 50 queries. Each query has a title field which we use in these experiments and refer to as the Very Short Query. Each query set also has corresponding relevance judgments for evaluation purposes.

Table 1 gives the statistics for query length both in stemmed and unstemmed forms for the collections we have used. Table 2 shows groups of queries in each collection based on their length after stopping. The complete query length statistics for all the collections can be seen [6].

TREC Topics	Ave query length (title field only)		Minimum query length		Maximum query length	
	non- stopped	stopped	non- stopped	stopped	non- stopped	stopped
301-350 (TREC-6)	2.7	2.5	1	1	5	4
351-400 (TREC-7)	2.5	2.4	1	1	3	3
401-450 (TREC-8)	2.5	2.4	1	1	4	3

Table 1: Query length statistics

Smart 13.0 is the retrieval system used for indexing, retrieval and evaluation in our work. Smart was developed at Cornell University and is an implementation of Salton's vector-space model. We use the *Lnu.ltu* weighting scheme which was developed by Singhal ([11], [12]). Chen [3] used some older weighting scheme for his work but Holtz [6] used this same scheme because it gives a much higher baseline in comparison to earlier methods and is considered to be a current standard for comparative evaluations.

	Query length after stopping			
	1	2	3	4
TREC-6	3	22	23	2
TREC-7	5	19	26	0
TREC-8	3	25	22	0
Merged	11	66	71	2

Table 2: Number of queries per collection binned on length after stopping

1.4 Holtz's Work

Holtz [6] concentrated his work on reranked pseudo-feedback. The algorithm for RPPF in Crouch *et. al.* [4] is outlined below:

For each query

Stage 1 (reranking)

1. To use M (e.g. 20) documents in the feedback process, retrieve N documents, where $N > M$.
2. For each document retrieved, compute a new similarity, Sim_{new} , of document to query, based on the reranking methods described below.
3. Rerank the N documents in decreasing value of Sim_{new} .

Stage 2 (pseudo-feedback)

1. Select terms from the top M documents in the reranked list and use them to expand the query.
2. Use the expanded query to retrieve the set of documents to be returned to the user.

Holtz [6] investigated three different reranking methods, one of which, CL, was developed by Chen:

1. *Coordination Level Reranking*(CL) : In this method the unstemmed terms in the query are compared to the unstemmed terms in each of the N documents retrieved. A value is assigned to each document, a count of the number of query terms matched. The documents are then reranked based on this value (and by the original similarity assigned by Smart within a specific coordination level).

2. *Intercoordination Level Reranking*(IC) : This method tries to improve over the CL algorithm. CL used the original similarity assigned by Smart for reranking intercoordination level documents.

Here intercoordination level documents were reranked using two and three word syntactic phrases found in the query.

3. Syntactic Phrase Reranking(SP) : Reranking is done here by assigning a new similarity based on single query terms, two word query phrases and three-word query phrases. Documents are reranked on the new similarity calculated without any regard to the coordination level.

For a detailed explanation of these three methods, see [3] and [6].

2. Initial Experiments

We started by investigating the effect of stemming on Holtz's method of reranking and reranked pseudo-feedback.

2.1 Using Unstemmed Terms and Phrases

Our initial set of experiments was to rerun Holtz's reranking, base pseudo-feedback and reranked pseudo-feedback algorithms using unstemmed query and document terms for reranking. Holtz used the `remove_s` stemmer for this purpose in his experiments. We wanted to analyze the difference between stemmed and unstemmed matching for the above mentioned methods (for detailed explanation of these algorithms refer to Holtz [6]).

Reranked pseudo-feedback produces a large improvement over base pseudo-feedback. The improvement can be attributed to the reranking of documents after the base retrieval. Reranking pulls more relevant documents to the top ranks so that better terms can be added to the query in the feedback run. We will investigate all three reranking methods: Coordination Level Reranking(CL), Intercoordination Level Reranking(IC), and Syntactic Phrase Reranking(SP). Holtz's experiments used unstemmed terms for reranking, with the exception of plurals and singulars. He used the `remove_s` stemmer which reduces plural terms to their singular form, but phrase matching was always done in the unstemmed form. Our experiments use no stemmer, for either terms or phrases.

CL reranking results will be observed more closely because it

reranks the documents based only on the single query terms without using phrases. So the difference in the results for CL will correspond directly to stemming. IC and SP reranking rerank based on both single query terms as well as two-term and three-term phrases formed from the query.

2.2 Parameters for the Feedback Run

Following the standard TREC procedure, we always retrieve 1000 documents in the base run.

1. M - the number of documents assumed relevant, is set to 20.
2. T - the number of terms by which query is expanded, is set to 300 as a result of earlier experiments[6].

In the Rocchio's feedback algorithm used for computing the feedback query :

3. α - the original query terms are reweighed using $\alpha = 8$.
4. β - the newly added terms are weighted using $\beta = 8$.
5. γ - for the negative feedback, is set to 8 (here γ down-weights the terms associated with the assumed non-relevant documents).
6. For the negative feedback, the number of documents assumed non-relevant, d , is set to 500.
7. N - the number of documents reranked, varies from 100 to 1000 in steps of 100.

The values for these parameters were established in Crouch *et. al.* [4] and Holtz [6].

2.3 Results

In these experiments we want to analyze the difference when using unstemmed terms and phrases over stemmed terms and phrases for reranking. Here we do not consider base pseudo-feedback

results as base pseudo feedback is independent of reranking and won't vary irrespective of whether reranking is performed with or without stemming.

Below are the tables showing Avg P and P@20 for the different reranking methods. N, the number of documents reranked, varies from 100 to 1000 and the best value of N is shown here. (For a complete listing for all values of N refer to the Tables 18-20.) We compare three experiments against baseline:

1. Simple reranking (RR)
2. Reranked pseudo-feedback with $\alpha = 8$, $\beta = 8$, $\gamma = 0$ (RRPF)
3. Reranked Pseudo Feedback with $\alpha = 8$, $\beta = 8$, $\gamma = 8$ (RRPF γ)

We evaluate these three experiments using Coordination Level Reranking(CL), Intercoordination Level Reranking(IC) and Syntactic Phrase Reranking(SP).

2.3.1 Reranking

Tables 3, 4 and 5 show the results for different reranking methods. It is hard to make a general statement for all the three methods, but by analyzing this data we can say that for all collections Avg P with unstemmed reranking is 1-3% lower than the stemmed cases. But unstemmed reranking showed an improvement of 1-2% in P@20 for Trec6 and Trec7. Unstemmed reranking doesn't help

Trec7 results at all (they are always 1-3% lower than the stemmed version). We don't see a big difference in the results.

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2240	13.94%	500	.2201	11.95%
Trec7	1000	.1733	11.30%	1000	.1701	9.25%
Trec8	900	.2179	10.95%	900,1000	.2171	10.54%
Base P@20(Trec6: .3180; Trec8: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	100	.3460	8.81%	100	.3510	10.38%
Trec7	200,1000	.3490	14.05%	400,1000	.3430	12.09%
Trec8	100	.3960	8.79%	200	.4000	9.89%

Table 3: CL Reranking (stemmed and unstemmed) compared to the baseline

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2204	12.11%	500	.2153	9.15%
Trec7	1000	.1671	7.32%	1000	.1639	5.27%
Trec8	900	.2124	8.15%	800	.2120	7.94%
Base P@20(Trec6: .3180; Trec8: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	500	.3420	7.55%	100	.3450	8.49%
Trec7	200	.3400	11.11%	100	.3310	8.17%
Trec8	100	.3900	7.14%	1000	.3930	7.97%

Table 4: SP Reranking (stemmed and unstemmed) compared to the baseline

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2252	14.55%	500	.2206	12.21%
Trec7	1000	.1746	12.14%	1000	.1708	9.70%
Trec8	900	.2146	9.27%	900	.2131	8.50%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	100	.3470	9.12%	100	.3510	10.38%
Trec7	600,800	.3480	13.73%	1000	.3390	10.78%
Trec8	100	.3910	7.42%	100	.3930	7.97%

Table 5: IC Reranking (stemmed and unstemmed) compared to the baseline

2.3.2 Reranked Pseudo Feedback (without Negative Feedback ($\gamma = 0$):

Evaluating the data below, we can say that the differences are not statistically significant. Since stemmed reranking sometimes provides better results than unstemmed, no general statement can be made about which one is better; the difference is never greater than 1-2.5%.

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	100	.2405	22.33%	300,500	.2403	22.23%
Trec7	1000	.2153	38.28%	1000	.2147	37.89%
Trec8	100	.2355	19.19%	100	.2357	20.01%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	400	.3630	14.15%	400	.3680	15.72%
Trec7	1000	.3920	28.10%	200,500	.3850	25.82%
Trec8	100	.4230	16.21%	500,600	.4200	15.38%

Table 6: RRPf with CL reranking (stemmed and unstemmed) compared against baseline; without negative feedback , $\alpha = 8$, $\beta = 8$, $\gamma = 0$

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2452	24.72%	500	.2469	25.58%
Trec7	100	.2086	33.98%	100	.2078	33.46%
Trec8	100	.2344	19.35%	100	.2359	20.11%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	500	.3780	18.87%	500	.3860	21.38%
Trec7	1000	.3860	26.14%	100	.3830	25.16%
Trec8	100	.4230	16.21%	100,200	.4180	14.84%

Table 7: RRPf with SP reranking (stemmed and unstemmed) compared against baseline; without negative feedback , $\alpha = 8$, $\beta = 8$, $\gamma = 0$

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	100	.2436	23.91%	600	.2424	23.80%
Trec7	100	.2117	35.97%	100	.2107	35.32%
Trec8	900	.2346	19.45%	100	.2354	19.86%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	700,800	.3750	17.92%	600,700,800	.3780	18.87%
Trec7	1000	.3910	27.78%	100	.3860	26.14%
Trec8	100	.4240	16.48%	200	.4200	15.38%

Table 8: RRRPF with IC reranking (stemmed and unstemmed) compared against baseline; without negative feedback , $\alpha = 8$, $\beta = 8$, $\gamma = 0$

2.3.3 Reranked Pseudo Feedback (with negative feedback, i.e., $\gamma = 8$):

Tables 9,10 and 10 show the final results for RRRPF with all three reranking methods. The same conclusion can be drawn about negative feedback as for non-negative feedback in the last section; the results are not that different so stemming is not a big issue here. One can choose either of the two but using unstemmed reranking may be advisable since stemming leads to some inaccuracies.

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	100	.2515	27.92%	800	.2510	27.67%
Trec7	1000	.2394	53.76%	1000	.2374	52.47%
Trec8	800	.2559	30.30%	100	.2553	29.99%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	100	.3740	17.61%	700	.3780	18.87%
Trec7	1000	.3950	29.08%	700	.3910	27.78%
Trec8	100	.4230	16.21%	600	.4210	15.66%

Table 9: RRPf_y with CL reranking (stemmed and unstemmed) compared against baseline; with negative feedback, $\alpha = 8$, $\beta = 8$, $\gamma = 8$

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2604	32.45%	800	.2590	31.74%
Trec7	200	.2314	48.62%	1000	.2298	47.59%
Trec8	100	.2562	30.45%	100	.2533	28.97%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	500	.3910	22.96%	900,1000	.3940	23.90%
Trec7	700	.3860	26.14%	700	.3860	26.14%
Trec8	100	.4230	16.21%	100,200	.4180	14.84%

Table 10: RRPf with SP reranking (stemmed and unstemmed) compared against baseline; with negative feedback, $\alpha = 8$, $\beta = 8$, $\gamma = 8$

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2591	31.79%	800	.2603	32.40%
Trec7	600	.2371	52.28%	1000	.2339	50.22%
Trec8	100	.2562	30.45%	1000	.2535	29.07%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Stemmed(with <i>remove_s</i>)			Unstemmed		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	900	.3910	22.96%	800,900	.3930	23.58%
Trec7	700,800	.3920	28.10%	700	.3880	26.80%
Trec8	100	.4260	17.03%	900	.4200	15.38%

Table 11: RRPf with IC reranking (stemmed and unstemmed) compared against baseline; without negative feedback , $\alpha = 8$, $\beta = 8$, $\gamma = 8$

2.4 Conclusion:

The final conclusion is that using unstemmed terms for reranking doesn't make a big difference in the RRPf results (both with and without negative feedback). Though we observed 1-2% improvement for P@20 in the reranking for Trec-6 and Trec7, no such improvement was reflected in RRPf results.

3. Phrase Selection

3.1 Motivation

Holtz [6] was able to achieve some nice improvement over the baseline with his Reranked Pseudo-Feedback. Three methods were used for the reranking phase: CL, SP and IC. Among these methods CL did not use any phrases, while SP and IC used syntactic phrases for reranking. Although RPPF gave good results for all three methods, RPPF results with IC and SP reranking were not any better than CL. In fact, SP reranked feedback performed worse than CL.

3.1.1 Holtz's Method of Phrase Selection

Holtz's[6] method of phrase selection was very simple and the phrases were chosen only from the queries. Only two and three-term phrases were used since the short queries consist of just two-three terms. Phrases were formed from unstemmed query terms after stop-listing. Two consecutively occurring terms formed a two word phrase and three consecutively occurring terms formed a three word phrase. With this scheme a query with L terms after stopping will have $L - 1$ two-term phrases and $L - 2$ three-term phrases.

3.1.2 Problem with Holtz's Method

This method used two-term and three-term phrases from the query for reranking. The problem was that after stopping, the queries only consisted of two-three terms. As seen from the Table 1, stopped query length is 2.5 for Trec6 and 2.4 for Trec7 and

Trec8. Therefore, the number of phrases associated with each query for reranking was very small.

We attribute the failure of SP- and IC-reranked pseudo-feedback (to make a significant improvement over CL-reranked pseudo feedback) to the lack of good phrases. Holtz was unable to determine with his methods difference between a *good phrase* and any other phrase. Our research concentrated on finding what we call *good phrases* for each query and then trying to get better results using SP reranking.

3.2 The N-gram Statistics Package (NSP)

We used a software package, the N-gram Statistics Package [10] developed by Pedersen and Banerjee, for finding phrases. They define an n-gram as "a sequence of 'n' tokens that occur within a window of at least 'n' tokens in the text." A token can be defined using a regular expression. We used two regular expressions to define tokens in our research:

1. `/w+['\.\@!_]*\w+/` (this matches any word having `'`, `.`, `@`, `!`, `_` as an interior character)
2. `/\w+/` (this matches any word)

A word in this context means any sequence of alpha-numeric characters. We used the 2-grams and 3-grams obtained using these regular expressions as two- and three-term phrases.

This package also has a set of statistical routines for ranking the phrases (or N-grams). These routines take a set of

phrases as input and rank them according to a measure calculated using their frequencies. Of the five available routines, we used Left Fisher's Test of Association because by observation it appeared to give us the best results (i.e., the results as evaluated by the researcher appeared to produce useful and meaningful phrases). This test we have used in our research is an exact test rather than an asymptotic one and Left Fisher's Test is just a shorthand for its complete name, left sided Fisher's Exact Test. (For a detailed explanation of NSP, see <http://www.d.umn.edu/~tpederse/nsp.html>)

3.3 Our Method of Phrase Selection

We observed from Holtz [6] that the short queries are too short to produce a sufficient number of phrases to improve the reranking. We would like to expand this set. Therefore, our method of phrase selection has two sources: (1) The query itself, (2) The top-ranked documents retrieved from the base run. A set of *good phrases* based on NSP was then formed for each query using the query text and the top-ranked documents returned for that particular query.

3.3.1 Phrases from the Query

All the phrases from the query are considered good and used for reranking. For each query the following algorithm was applied to the query text to form phrases:

1. Two and three-term phrases were formed from the original query text by NSP using the above mentioned regular expressions. NSP produces a few extra phrases from the query as compared to Holtz's method. This was because we used a window of two to create the phrases. For example, if the query is '*international organized crime*', then NSP gives the following three two-term phrases: '*international organized*', '*organized crime*' and '*international crime*'. Here '*international crime*' is the new phrase added. Windowing did not make any difference for three-term phrases since the length for a query should be at least 4 to introduce an extra three-term phrase.
2. In this step the list of phrases obtained from Step 1 is refined. Each term contained in the phrase is examined separately and any phrase which contains a term matching a common word or formed with digits is removed.
3. The list obtained from Step 2 is considered to be the set of *good phrases* for that particular query.

3.3.2 Phrases from Top-ranked Documents

Each test collection contains more than 500,000 documents. We had to choose whether to consider all of these documents and make a static list of *good phrases* (a phrase dictionary for the collection) or to create a set of phrases for each query. We chose the latter approach. After a base retrieval we used the top-ranked documents to extract the *good phrases* for that particular query.

The phrase selection method varies slightly depending on whether relevance information from top-ranked documents is or is not used. The steps in the algorithm are same in both the cases except for the first step.

3.3.2.1 Experiments with Relevance Feedback:

To generate the phrases:

1. From the **Top n** (1000) documents retrieved, **Top R** (5 or 10) documents are selected. The relevant documents among these **Top R** documents are merged and NSP is applied to yield the *good phrases*. The parameters for NSP are set so that a phrase must have occurred at least twice to be considered a *good phrase*. (The non-relevant documents among **Top R** can also be merged to find the bad phrases.)
2. The list obtained from Step 1 is refined as explained in the previous section (Step 2 in Section 3.3.1) by removing any phrases consisting of digits or common words.
3. Since the documents are very long in comparison with the queries, this list can be really long. So we applied the statistical routine (Left Fisher's Test of Association) of NSP to get a ranked list of these phrases for each query. Many phrases are produced at the same rank. To reduce the size of the phrase set, we consider only phrases ranked 1-5 (This produces a variable-size set, ranging from 40 to 200 phrases).
4. After careful examination of the list, it was observed that we were getting reasonable phrases at the top of the list but this is not what we are really interested in. We are interested in phrases which are *good* (i.e. hold more meaning than the individual terms forming the phrase) and also represent the document they are contained in. For example: *permanent infrastructures* might be a very good phrase on its own, but for the query '*international organized crime*' phrases like '*crime ring*' and '*crime groups*' seem to be more useful. So, if we are trying to find good phrases for this query, '*crime ring*' and '*crime groups*' are considered better phrases than '*permanent infrastructures*' although they are lower in rank. Considering this, we rerank the list obtained from Step 3. (This phrase reranking is explained in Appendix.)
5. Reranking is done in Step 4 because the list obtained from Step 3 was still long and only a few phrases can be selected as *good* from the list. Finally, we chose the Top d (5, 10, 15 or 20) phrases as the *good phrases* for a particular query.

3.3.2.2 Experiments with Pseudo Feedback

To generate *good phrases* for pseudo feedback experiments:

1. From the **Top n** (1000) documents retrieved, The **Top PD** (PD=5,10,15 or 20) documents are merged and NSP is applied to find the *good phrases* without any consideration to whether or not they come from relevant documents.

The remaining steps (Steps 2, 3, 4 and 5) are same as in the previous section for relevance feedback experiments.

4. Experiments with Relevance Feedback

As explained in Section 3.3, a major problem identified with Holtz's RRPF was that the number of phrases used in reranking was too small to make a significant improvement over non-phrase reranking methods. One approach is to generate more *good phrases* from the top ranked relevant documents. This led us to relevance feedback.

We performed two different experiments in an attempt to improve upon the regular relevance feedback. First, we review relevance feedback and then explain our experiments.

4.1 Relevance Feedback

In relevance feedback, a retrieval using the original query vector is performed. New terms chosen from top relevant documents are added to the query vector. The evaluation for relevance feedback is very different from the evaluation used for pseudo-feedback. During this evaluation, the documents retrieved in the first iteration are discarded. In other words, the second retrieval is done on the residual collection. This type of evaluation is necessary to judge the real effectiveness of relevance feedback, because the new query is formed using the relevance information from the top-ranked documents, and if these top-ranked documents are considered in the evaluation process it would bias the results.

Table 12 shows P@20 and Avg P for relevance feedback with different values of R (the number of top-ranked documents considered as the source for relevance information). The standard values are used for the parameters:

1. T - the number of terms added to the query vector, is set to 300 [4].
2. *Lnu.ltu* weighting is used [6].
3. $\alpha = 8$, $\beta = 8$.

In the following table, relevance feedback is compared to the baseline.

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Average Precision			%Increase over base		
	R = 5	R = 10	R = 20	R = 5	R = 10	R = 20
Trec6	.1840	.1816	.1688	-6.4%	-7.63%	-14.14%
Trec7	.1833	.1893	.1804	17.73%	21.58%	15.86%
Trec8	.1827	.1740	.1470	-6.98%	-11.41%	-25.15%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	P@20			%Increase over base		
	R = 5	R = 10	R = 20	R = 5	M = R	R = 20
Trec6	.3770	.3710	.3590	18.55%	16.67%	12.89%
Trec7	.3650	.3650	.3970	19.28%	19.28%	29.74%
Trec8	.3820	.3870	.3600	4.95%	6.32%	-1.10%

Table 12: Relevance Feedback with varying R (the number of documents used for relevance information), $\alpha = 8$, $\beta = 8$. Evaluation done using the residual collection

From this table, we can say that for two collections relevance feedback does not improve Avg P. Avg P goes down for all cases except Trec7. We also notice that looking at a larger number

of documents does not necessarily help. Avg P and P@20 with R = 5 and R = 10 are better most of the time than at R = 20.

Consider Avg P. The relevance feedback values are low and the percentages at R = 20 are the worst in two cases (Trec6 and Trec8). These results can be partially attributed to the way relevance feedback is evaluated. Since the documents considered for relevance information are removed from the evaluation process, it maybe better to consider fewer documents. Our general aim should be to improve retrieval by looking at fewer documents and extracting the maximum information from them. We will use this approach in our next set of experiments.

4.2 Hybrid Feedback

Our first experiment to improve upon *relevance feedback* is a hybrid of relevance feedback and pseudo feedback. As Table 12 reveals, it may be better to look at fewer documents for relevance information. This leads us to investigate a method which uses relevance information from some documents and pseudo-relevance from others.

Base Pseudo-Feedback: In base pseudo-feedback, we do a base retrieval and then assume the **Top M** documents to be relevant. We add terms from these documents to the query vector via Rocchio's method. Since the top-ranked documents used for query reformulation are only *assumed* to be relevant and their real relevance judgments are not seen, they can be considered in the

evaluation (i.e., there is no need for a feedback on the residual collection).

The algorithm for hybrid feedback is outlined below. The steps are subsequently explained in more detail.

1. A base retrieval is done and **n** ($n=1000$) documents are retrieved.
2. A total of **M** documents (10 or 20) are used for expansion of query.
3. Relevance judgments are seen for **Top R** documents (half of the set **M**, 5 or 10).
4. The relevant documents among the **Top R** are used to select good phrases and the non-relevant documents among the **Top R** are used to select bad phrases.
5. The **n - R** documents are reranked using Syntactic Phrase reranking (SP) based on the phrases selected above.
6. The **Top M - R** (5 or 10) documents from the reranked list (of Step 4) are assumed to be relevant and used for query reformulation.
7. Finally, the query is reformulated using terms from the relevant documents in the **Top R** and the **Top M - R** documents from the reranked list. (Hence, we can also say that relevance information was used from the **Top R** documents obtained from the base retrieval and pseudo relevance information was used from the **Top M - R** documents of the reranked list.)
8. The **Top R** documents have been seen, so they are not considered for evaluation.

Our main aim in reranking the **n - R** documents is to get as many relevant documents into the top ranks as possible and thus improve the feedback query.

Details of the experiment follow.

4.2.1 Phrase Selection

In this experiment, we use Smart to produce phrases (i.e., we index the whole of the collection with two-word phrases as a

separate concept type). Here we did not use the NSP regular expressions to find phrases and only two word phrases were considered. Smart indexes the collection with phrases by considering any two adjacent terms as a phrase and giving the frequency of all such phrases. The statistical routine in NSP was used to distinguish between phrases (i.e., to differentiate good phrases from bad ones) and produce a ranked list of phrases.

So the good phrases here come from two sources:

1. The query - Queries are indexed with phrases as a separate type. All two-word phrases from the query are considered *good*.
2. The top-ranked documents - Phrases here come from the relevant documents (in Top 5 or 10). The top 20 phrases from this list are considered *good* and used for reranking.

We also selected some bad phrases from the non-relevant documents in the **Top R** documents. A ranked list of bad phrases is formed in the same way and Top 20 phrases in this list are considered bad for the purpose of these experiments. The value of 20 for good and bad phrases was finalized by observation and some experimentation.

At this point we have three sets of phrases: A, B and C. Set A consists of two-word phrases from the query, B consists of two-word phrases from the relevant documents in the **Top R** documents, and C consists of bad phrases from the non-relevant documents in the **Top R**.

4.2.2 Reranking

One goal of this work was to improve upon the Syntactic Phrase Reranking(SP) method of Holtz [6]. This is a simple method and with better phrases, we might expect it to give better results.

SP calculates a new similarity value for each query-document pair based on the terms and phrases associated with the query (as in Section 4.2.1) and then reranks them by their new similarity. The similarity of each query-document pair is calculated by the following formula:

$$D_{sim} = t + (3 * a) + (2 * b) - (2 * c)$$

Where, t = number of unique query terms present in the document

a = number of Set A phrases present in the document

b = number of Set B phrases present in the document

c = number of Set C phrases present in the document

Ties are broken by the original weighting given to the documents by Smart. Experiments were performed using only phrases from Set A and Set B as well, but the results are better for Trec6 and Trec7 when bad phrases (i.e., Set C phrases) are not considered.

4.2.3 Results

We tested our method for two different values of M (the number of documents used for query expansion), at 10 and 20. When

$M = 10$, the top 5 documents of the base retrieval are used for relevance feedback (and for finding the phrases) and the remaining 5 are assumed relevant and are the top 5 of the reranked list. Similarly, when $M = 20$, the top 10 documents of the base retrieval are used for relevance feedback and for phrases, and remaining 10 are assumed relevant. In both the cases, the documents used for relevance information, i.e., the top 5 and top 10, respectively, are not considered in the evaluation. Here we show the best results, in this case for $M = 10$. For complete results refer to the Tables 21-22.

The other parameters have the same values as suggested in Crouch et al. [4].

1. n - always 1000 documents are retrieved in the base run.
2. T , the number of terms by which query is expanded, is set to 300.
3. $\alpha = 8$.
4. $\beta = 8$.
5. N , the number of documents reranked, varies from 100 to 1000 in steps of 100.

Base Average Precision(Trec6: .1966;Trec7:.1557; Trec8: .1964)						
Collect ion	Rel. Feedback		Hybrid Feedback(with <i>bad phrases</i>)			
	Avg. P.	%Inc. over base	N*	Avg.P.	%Inc. over base	%Inc over Rel. Fdk.
Trec6	.1840	-6.41%	400	.1942	-1.22%	5.54%
Trec7	.1833	17.73%	900	.1995	28.13%	8.84%
Trec8	.1827	-6.98%	400	.1830	-6.82%	.16%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collect ion	Rel. Feedback		Hybrid Feedback(with <i>bad phrases</i>)			
	P@20	%Inc. over base	N*	P@20	%Inc. over base	%Inc over Rel. Fdk.
Trec6	.3770	18.55%	200	.3740	17.61%	-.80%
Trec7	.3650	19.28%	700	.4010	31.05%	9.86%
Trec8	.3820	4.95%	400	.3760	3.30%	-1.57%

**Table 13a: Comparison of Relevance Feedback and Hybrid Feedback(with bad phrases); M = 10, R = 5, $\alpha = 8$, $\beta = 8$,
* When we use R relevant documents for reranking, N = N - R**

These results show that the best case Avg P with our method always improves over the normal Relevance Feedback. The best case P@20 with our method is always better than the base case P@20. Although it produces an improvement of roughly 10% over normal relevance feedback for Trec7, the results are equivalent to Holtz [6] for Trec6 and Trec8.

Now let us consider the results for the experiments when the *bad phrases* (i.e., Set C phrases) are not considered. We show the best results in Table 13b (for a complete reference see Tables 23-

24). Table 13b shows that for two of the collections (Trec6 and Trec7), our method using only the *good phrases* performs much better

Base Average Precision(Trec6: .1966;Trec7:.1557; Trec8: .1964)						
Collect ion	Rel. Feedback		Hybrid Feedback(no <i>bad phrases</i>)			
	Avg. P.	%Inc. over base	N*	Avg.P.	%Inc. over base	%Inc over Rel. Fdk.
Trec6	.1840	-6.41%	300	.2069	5.24%	12.45%
Trec7	.1833	17.73%	300	.2018	29.61%	10.09%
Trec8	.1827	-6.98%	100	.1780	-9.37%	-2.57%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collect ion	Rel. Feedback		Hybrid Feedback(no <i>bad phrases</i>)			
	P@20	%Inc. over base	N*	P@20	%Inc. over base	%Inc over Rel. Fdk.
Trec6	.3770	18.55%	300	.3840	20.75%	1.86%
Trec7	.3650	19.28%	200	.3830	25.16%	4.93%
Trec8	.3820	4.95%	400	.3640	0%	-4.71%

Table 13b: Comparison of Relevance Feedback and Hybrid Feedback (without bad phrases); M = 10, R = 5, $\alpha = 8$, $\beta = 8$

* When we use R relevant documents for reranking, $N = N - R$

than the regular feedback. It improves upon both the Avg P and the P@20.

4.2.4 Conclusion

We conclude here that when comparing hybrid feedback (without bad phrases) to relevance feedback, our method gives better results for Avg P. For P@20, results not worse than the normal relevance feedback and in the case of Trec7 it is substantially better. For Trec6 and Trec7, the method using only good phrases is better than the one with both good and bad phrases.

4.3 Retrieval with Phrases Included in Query Vector

In all of the previous experiments, phrases were used only for reranking purpose (i.e., not in the reformulated query). Now we consider this possibility. We retrieve with query vectors which contain both single word terms and two word phrases. This requires the documents as well as the queries to be indexed with phrases. *Lnu.ltu* weighting is used as in all other experiments. Base retrieval with this method is quite simple. Both the Smart-generated phrases and terms are given *ltu* weighting and the retrieval is performed.

Table 14 compares the base retrieval done both with and without phrases included in the query vector (the two-word phrases are generated by Smart and come from the queries only).

Collection	Base Ret. without phrases		Base Ret. with phrases	
	Avg. P.	P@20	Avg. P.	P@20
Trec6	.1966	.3180	.1719	.2650
Trec7	.1557	.3060	.1466	.3100
Trec8	.1964	.3640	.1887	.3430

Table 14: Avg P and P@20 for Base Retrievals with and without phrases included in the query vector

As it can be seen from this table, the retrieval with phrases included in the query vector performs badly. This was not a good sign. Next we did normal relevance feedback with phrases by expanding the original query with 300 single word terms and 20 two

word phrases. But again the results were far below the normal relevance feedback using only single word terms. We did not experiment further with this approach. Using our method and Smart-generated phrases, we can't construct a better query than the original. Thus reranking may prove more effective.

5. Reranking and Reranked Pseudo-Feedback with Assumed Relevance for Phrase Selection

Our attempts to improve upon relevance feedback were not very successful. Relevance feedback also involves user intervention. So again we turn to pseudo feedback and selecting phrases without using relevance information.

Our basic algorithm is the same, with Stage 1 as reranking and Stage 2 pseudo-feedback (as in [6]). The only difference is the way phrases are selected and the method used for reranking.

5.1 Phrase Selection

The phrase selection method here is based on NSP (exactly as described in Section 3.3.2.2). The only decision to be made is the number of top ranked documents used to generate the phrases. We experiment by using documents in multiples of 5 (i.e., starting with the 5 top-ranked documents, then 10). The results were worse as we used more than 5 documents for phrase selection, so we finalized the number of documents at 5.

After applying the algorithm (see Section 3.3.2.2), we have two ranked list of phrases - one of two word phrases, and the other of three-word phrases. The next question is how many phrases of each type to select for reranking. We experiment with 5, 10, 15 and 20 two-word phrases (and the same for three-word phrases). These phrases (from the top 5 documents) are by definition considered to be *good*. The phrases selected can be divided into four categories:

1. Two-word phrases from queries (Set A)
2. Three-word phrases from queries (Set B)
3. Two-word phrases from the top-ranked documents (Set C)
4. Three-word phrases from the top-ranked documents (Set D)

5.2 Reranking

Reranking is done using a modified version of Syntactic Phrase Reranking (SP) very similar to the one in Section 4.2.2. A new similarity is calculated for each document retrieved, based on the phrases, and then the documents are reranked. No special consideration is given to the number of times a particular phrase occurs in a document. If a phrase occurs in a document, a point value is added to the similarity for occurrence of the phrase. Below is the formula used to calculate the similarity:

$$D_{sim} = (2 * t) + (3 * a) + (4 * b) + (2 * c) + (2 * d)$$

Where, t = number of unique query terms present in the document

a = number of Set A phrases present in the document

b = number of Set B phrases present in the document

c = number of Set C phrases present in the document

d = number of Set D phrases present in the document

Sets A, B, C and D refer to the sets of phrases described in the previous section.

5.3 Results

Reranking and RPPF without negative feedback results are discussed here. (We also did some experiments for RPPF with

negative feedback but the results were not encouraging; c.f.r. Table 28.)

The parameters have the same values as used earlier and suggested in Crouch, *et. al.*[4].

1. Always 1000 documents are retrieved in the base run.
2. M - the number of documents assumed relevant for the feedback run, is set to 20.
3. T - the number of terms by which query is expanded, is set to 300.
4. $\alpha = 8$
5. $\beta = 8$.
6. N - the number of documents reranked, will vary from 100 to 1000 in steps of 100.

There are three additional parameters here:

1. PD - the number of documents used for phrase selection, set to 5 by experimentation.
2. $P2$ - the number of two word phrases picked up from top 5 documents and considered good ($P2 = 5, 10, 15, 20$).
3. $P3$ - the number of three word phrases picked up from top 5 documents and considered good ($P3 = 5, 10, 15, 20$).

All the two-word and three-word phrases generated from queries are considered *good* and used in reranking so there is no parameter governing their number.

5.3.1 Reranking

We compare our reranking results with those of Holtz [6] and with the baseline. Our method of phrase selection produces more phrases than Holtz's [6] method. The only parameter that we need to finalize is the number of two- and three word phrases that

are to be used for reranking. For two out of three collections, namely, Trec6 and Trec7, we get best reranking results when 5 two word phrases and no three word phrases are used. For Trec8, the best result is obtained with 5 two-word phrases and 5 three-word phrases. (The result with 5 two-word phrases produces P@20 that is approximately 6% lower.) Table 15 shows the best results for reranking with 5 two-word phrases. (For a complete listing of results refer Tables 25-27.)

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Holtz's Reranking			Our Reranking(with NSP)		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2204	12.11%	500	.2177	10.73%
Trec7	1000	.1671	7.32%	1000	.1845	18.50%
Trec8	900	.2124	8.15%	400	.2141	9.01%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Holtz's Reranking			Our Reranking(with NSP)		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	500	.3420	7.55%	900	.3680	15.72%
Trec7	200	.3400	11.11%	1000	.3640	18.95%
Trec8	100	.3900	7.14%	100	.3790	4.12%

Table 15: Holtz's SP Reranking and our SP reranking(with 5 two word phrases coming from Top 5 documents) compared to the baseline

As we see from Table 15, our reranking scheme produces a nice improvement throughout (except for P@20 for Trec8 and Avg P for Trec6 - the Avg P for Trec6 is roughly 1% down and can be considered pretty much flat). Also, P@20 for our method beats the

P@20 for Holtz's Intercoordination Level(IC) and Coordination Level Reranking which performed much better than his SP reranking. Hence we can conclude that our reranking scheme (with 5 two-word phrases from the Top 5 documents) can be safely used over that of Holtz's.

5.3.2 Reranked Pseudo-Feedback

Reranking gives us good results throughout, so theoretically Reranked Pseudo-Feedback should also perform well. Unexpectedly, we did not get uniformly good results for RPPF. We got a significant improvement for Trec7 but results went down for Trec6 and Trec8.

Table 16 shows the best values for RPPF. P2 and P3 are two-word and three-word phrases, respectively, used for reranking. We improve P@20 for Trec7 by approximately 5% and Avg P by 7%. For Trec6 it was pretty much flat but in Trec8 our results were worse.

Regarding the number of phrases: we conclude that P3 = 5 gives good results for two collections and P3 = 0 for one, so the number of three word phrases shouldn't be more than 5. P2 = 5 was good for two of the collections and 10 for one. We can conclude it is not advisable to go beyond 10 phrases.

Base Average Precision(Trec6: .1966; Trec7: .1557; Trec8: .1964)						
Collection	Holtz's RRPf			Our RRPf		
	Best N	Avg. P	%Inc	Best N	Avg. P	%Inc
Trec6	500	.2452	24.72%	P2 = 10 ,P3 = 5		
	300	.2403				22.23%
Trec7	100	.2086	33.98%	P2 = 5, P3 = 5		
	900	.2196				41.04%
Trec8	100	.2344	19.35%	P2 = 5, P3 = 0		
	100	.2228				13.44%
Base P@20(Trec6: .3180; Trec7: .3060; Trec8: .3640)						
Collection	Holtz's RRPf			Our RRPf		
	Best N	P@20	%Inc	Best N	P@20	%Inc
Trec6	500	.3780	18.87%	P2 = 10, P3 = 5		
	400	.3680				15.72%
Trec7	1000	.3860	26.14%	P2 = 5, P3 = 5		
	300	.4010				31.05%
Trec8	100	.4230	16.21%	P2 = 5, P3 = 0		
	400	.3880				6.59%

Table 16: RRPf with SP reranking (Holtz's and NSP) compared against baseline; without negative feedback, $\alpha = 8$, $\beta = 8$, $\gamma = 0$

In the end, we conclude that our simple reranking scheme (Table 15) is very successful and RRPf gives results comparable to those of Holtz [6] for Trec6 and Trec7.

6. Conclusions and Suggestions for Future Work

The experiments performed in this thesis can be divided into two categories, one involving the use of relevance information and the other using pseudo relevance. We summarize the results for these two categories separately.

6.1 Experiments with Relevance Feedback

Two different kind of hybrid feedback experiments were performed: (1) using both *good* and *bad phrases* for reranking purposes (we refer to it here as HFb), and (2) using *good phrases* only (which we refer to as HF). Best case results for Trec6 and Trec7 make an improvement over the regular relevance feedback, though HF gives slightly better results than HFb. The number of documents used for query reformulation, M , is set to 10 and the number of documents used for relevance information, R , is set to 5. These parameters give the best results and are uniform across the collections. The reranking parameter, N , is set to 300 for HF and 400 for HFb. Table 17 compares the hybrid feedback with relevance feedback and pseudo feedback (the results shown are for the parameter values mentioned above and are close to the best results).

6.2 Experiments with Pseudo-Feedback

We tried to improve upon RPPF as shown in Holtz [6], the difference being in the phrases used for reranking. For our experiments, phrases are selected using NSP. Although we were,

Trec6 (base P@20 = .3180, base Avg P = .1966)			
	RF(R=20)	HF(N=300)	HFb(N=400)
Avg P	.1688	.2069	.1942
P@20	.3590	.3840	.3750
Trec7 (base P@20 = .3060, base Avg P = .1557)			
	RF(R=20)	HF(N=300)	HFb(N=400)
Avg P	.1893	.2018	.1971
P@20	.3970	.3800	.3920
Trec8 (base P@20 = .3640, base Avg P = .1964)			
	RF(R=20)	HF(N=300)	HFb(N=400)
Avg P	.1740	.1749	.1830
P@20	.3600	.3630	.3760

Table 17: Comparison of RF, HF and HFb (Standard parameters for feedback as in [4], $T=300$, $\alpha=8$, $\beta=8$)

Successful in improving the reranking over that of [6], that improvement is not reflected in the feedback runs. Table 18 compares our reranking and RRPf with base pseudo-feedback and relevance feedback. We have been able to select collection-independent values for our reranking and RRPf experiments. Although the results might not be the best with this parameter set, they are very close to the best and give good results overall. These parameters settings are:

1. PD - number of documents used to select phrases, is set to 5.

2. P2 - number of two-word phrases used in reranking, is set to 5.
3. P3 - the number of three-word phrases used in reranking, is set to 0.
4. N - the number of documents reranked, is set to 800.

Trec6 (base P@20 = .3180, base Avg P = .1966)				
	PF	RF(R=20)	RR	RRPF
Avg P	.2201	.1688	.2169	.2278
P@20	.3380	.3590	.3660	.3670
Trec7 (base P@20 = .3060, base Avg P = .1557)				
	PF	RF(R=20)	RR	RRPF
Avg P	.1988	.1893	.1832	.2192
P@20	.3510	.3970	.3640	.3980
Trec8 (base P@20 = .3640, base Avg P = .1964)				
	PF	RF(R=20)	RR	RRPF
Avg P	.2264	.1740	.2142	.2180
P@20	.4010	.3600	.3730	.3860

Table 18: Comparison of PF, RF, RR and RRPf (Standard parameters for feedback as in [4], T=300, $\alpha=8$, $\beta=8$, N=800)

From Table 17 we see that RR (NSP based reranking), which does not involve reformulation of the query, gives results as good as or better than Relevance Feedback almost all the time (except P@20 for Trec7). Considering P@20, RR also performs better than PF for Trec6 and Trec7. So, RR is quite effective considering the

fact that results comparable to PF and RF are being obtained without incurring the cost of query reformulation and a feedback run.

6.3 Future Work

In our experiments, phrases do not produce a significant improvement over the methods not based on phrases. In our experiments, only Syntactic Phrase Reranking was used with the NSP selected phrases. One suggestion is to try the Intercoordination Level Reranking method in Holtz [6] and test it with the same phrase set.

For a collection with a higher than average baseline, like Trec8, relevance feedback gives a lower result for both Avg P and P@20 (i.e., the original query performs so well that relevance feedback is not able to improve upon it). Yet NSP-based phrase reranking performs well in these circumstances. (It also performs better than Holtz's reranking for Trec6 and Trec7.) In similar cases, one might like to consider using reranking in lieu of relevance feedback.

7. References

- [1] N.J.Belkin, C.Cool, J.Head, J.Jeng, D.Kelly, S.Lin, L.Lobash, S.Y.Park, P.Savage-Knepshield, C.Sikora. Relevance Feedback versus Local Context Analysis as Term Suggestion Devices: Rutgers' TREC-8 Interactive Track Experience. In *Proceedings of the Eighth Text Retrieval Conference(TREC-8), Gaithersburg, Maryland,1999.*
- [2] C.Buckley (1996). Smart people mail archive. <ftp://ftp.cs.cornell.edu/pub/smart>
- [3] Q.Chen. Improving the Retrieval Effectiveness of Very Short Queries. M.S. Thesis, University of Minnesota Duluth,1999.
- [4] C.Crouch, D.Crouch, Q.Chen & S.Holtz. Improving the Retrieval Effectiveness of Very Short Queries. *Information Processing and Management,Vol.38,Number 1,2002.*
- [5] F.C.Gey and A.Chen. Phrase Discovery for English and Cross-language Retrieval at TREC-6. In *Proceedings of the Sixth Text Retrieval Conference(TREC-6), Gaithersburg, Maryland, 1997.*
- [6] S.Holtz. Further Experiments in Improving Very Short Queries. M.S. Thesis, University of Minnesota Duluth,2001.
- [7] W.Kraaij and R.Pohlmann. Comparing the Effect of Syntactic vs. Statistical Phrase Indexing Strategies for Dutch. From <http://citeseer.nj.nec.com>
- [8] M.Mitra, A.Singhal and C.Buckley. Improving Automatic Query Expansion. In *Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval, Melbourne, Australia,1998.*
- [9] M.Mitra, C.Buckley, A.Singhal and C.Cardie (1997). An Analysis of Statistical and Syntactic Phrases. In *Proceedings of RIAO-97, Montreal, Canada, 1997.*
- [10] T. Pedersen & S. Banerjee (2002). N-gram Statistics Package, version 0.51. From <http://www.d.umn.edu/~tpederse/nsp.html>
- [11] A.Singhal. AT&T at Trec6. In *Proceedings of the Sixth Text Retrieval Conference(TREC-6), Gaithersburg, Maryland, 1997.*
- [12] A.Singhal, J.Choi, D.Hindle, D.D.Lewis and F.Pereira (1998). AT&T at TREC-7. In *Proceedings of the Seventh Text Retrieval Conference(TREC-7), Gaithersburg, Maryland,1998.*
- [13] E.M.Voorhees and D.Harman. Overview of the Eighth Text Retrieval Conference(TREC-8). In *Proceedings of the Eighth Text Retrieval Conference(TREC-8), Gaithersburg, Maryland, 1999.*
- [14] C.Zhai (1997). Fast Statistical Parsing of Noun Phrases for Document Indexing. From <http://citeseer.nj.nec.com>

8. Appendix

8.1 Reranking of the Phrase List given by NSP

The final list of phrases obtained using NSP (Step 3 of algorithm described in Section 3.3.2.1) usually consists of 40-200 phrases. For our experiments, the number of phrases considered *good* for each query varies from 5-20. It is also observed that many meaningful phrases (which directly relate to the query) occur down in the list. Seeing this, we decided to rerank the list obtained to percolate these *good phrases* to the top of the list.

Before explaining the reranking scheme, we first describe the output format of the list obtained from NSP. Following is the list of phrases obtained for the query 'New Hydroelectric Projects':

```
1273
billion<>yuan<>1 1.0000 2 3 4
electric<>power<>1 1.0000 2 2 17
ertan<>hydroelectric<>1 1.0000 4 5 24
power<>stations<>1 1.0000 2 17 3
feasibility<>studies<>1 1.0000 3 3 3
solar<>electric<>1 1.0000 2 2 2
river<>successfully<>1 1.0000 2 6 3
ferc<>staff<>1 1.0000 2 4 3
hydroelectric<>power<>1 1.0000 11 24 17
power<>station<>1 1.0000 6 17 11
hydroelectric<>station<>1 1.0000 4 24 11
million<>yuan<>1 1.0000 2 4 4
article<>type<>1 1.0000 2 3 2
current<>year<>1 1.0000 2 2 4
```

```

station<>projecting<>1 1.0000 2 11 2
nam<>ngum<>1 1.0000 3 3 3
travel<>approximately<>1 1.0000 2 2 2
hydroelectric<>plants<>1 1.0000 2 24 4
successfully<>blocked<>1 1.0000 3 3 3
site<>visit<>1 1.0000 6 9 6
hydroelectric<>project<>2 0.9999 3 24 13
hydroelectric<>projects<>3 0.9997 2 24 8

```

1273 is the total number of phrases obtained before refining the list. The number following the phrase is its rank (1 in case of phrase '*electric power*'), the second number is the measure by which phrases are ranked (1.0000 in case of '*electric power*'), the third is the number of times that phrase occurred in the text (2), fourth is the number of times the first term of the phrase ('*electric*') occurred as the first term of any phrase in the text, and the last number (17) denotes the number of times the second term of phrase ('*power*') occurs as the second term of any phrase in the text.

It can be seen that the phrases we are interested in, namely, 'hydroelectric projects', 'hydroelectric plants', 'power station', 'hydroelectric station,' occur down in the list.

Thus we rerank by adding the last three numbers for each phrase and rerank the list based on the sum. Ties are broken using the original ranking. It is observed that good phrases tend to

concentrate at the top with this scheme. Below is the reranked list for the above example:

1273

hydroelectric<>power<>1 1.0000 11 24 17
hydroelectric<>project<>2 0.9999 3 24 13
hydroelectric<>station<>1 1.0000 4 24 11
power<>station<>1 1.0000 6 17 11
hydroelectric<>projects<>3 0.9997 2 24 8
ertan<>hydroelectric<>1 1.0000 4 5 24
hydroelectric<>plants<>1 1.0000 2 24 4
power<>stations<>1 1.0000 2 17 3
electric<>power<>1 1.0000 2 2 17
site<>visit<>1 1.0000 6 9 6
station<>projecting<>1 1.0000 2 11 2
river<>successfully<>1 1.0000 2 6 3
million<>yuan<>1 1.0000 2 4 4
ferc<>staff<>1 1.0000 2 4 3
successfully<>blocked<>1 1.0000 3 3 3
feasibility<>studies<>1 1.0000 3 3 3
billion<>yuan<>1 1.0000 2 3 4
nam<>ngum<>1 1.0000 3 3 3
current<>year<>1 1.0000 2 2 4
article<>type<>1 1.0000 2 3 2
travel<>approximately<>1 1.0000 2 2 2
solar<>electric<>1 1.0000 2 2 2

Here is another example showing the effectiveness of this reranking scheme. The top 20 phrases in the list before and after reranking are shown.

Before Reranking	After Reranking
Federation president	Organized crime
Permanent infrastructures	International crime
Increasing control	Fight crime
Conference views	Crime groups
Europe warnings	Crime text
Significant organized	Italian crime
Janusz wolny	European crime
Italian groups	Crime group
Eastern germany	Documented crime
Dresden conference	Crime rings
typebfn	Crime bosses
Favorite place	International cooperation
Methods similar	Text international
Western european	Criminal groups
Crime text	Czech republic
Security council	Countries organized
Increasing quantities	Interior minister
Organized crime	International conference
Virtually impossible	Czech criminal
Federation president	Italian groups

Table 19: Top 20 phrases before and after reranking for the query - 'international organized crime'

8.2 Results

Results follow in Tables 20-36.

Trec6												
N	Avg P(baseline .1966)						P@20(baseline .3180)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2105	7.07%	.2063	4.93%	.2109	7.27%	.3510	10.38%	.3450	8.49%	.3510	10.38%
200	.2134	8.55%	.2074	5.49%	.2123	7.99%	.3460	8.81%	.3390	6.60%	.3480	9.43%
300	.2200	11.90%	.2139	8.80%	.2194	11.60%	.3440	8.18%	.3350	5.35%	.3450	8.49%
400	.2194	11.60%	.2140	8.85%	.2195	11.65%	.3440	8.18%	.3370	5.97%	.3460	8.81%
500	.2201	11.95%	.2153	9.51%	.2206	12.21%	.3430	7.86%	.3420	7.55%	.3490	9.75%
600	.2198	11.80%	.2150	9.36%	.2203	12.05%	.3410	7.23%	.3370	5.97%	.3470	9.12%
700	.2193	11.55%	.2142	8.95%	.2191	11.44%	.3400	6.92%	.3350	5.35%	.3460	8.81%
800	.2191	11.44%	.2142	8.95%	.2190	11.39%	.3400	6.92%	.3350	5.35%	.3460	8.81%
900	.2188	11.29%	.2138	8.75%	.2185	11.14%	.3380	6.29%	.3330	4.72%	.3440	8.18%
1000	.2186	11.19%	.2143	9.00%	.2187	11.24%	.3380	6.29%	.3330	4.72%	.3450	8.49%
Trec7												
N	Avg P(baseline .1557)						P@20(baseline .3060)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.1662	6.74%	.1595	2.44%	.1648	5.84%	.3410	11.44%	.3310	8.17%	.3370	10.13%
200	.1677	7.71%	.1607	3.21%	.1666	7.00%	.3370	10.13%	.3280	7.19%	.3340	9.15%
300	.1690	8.54%	.1603	2.95%	.1673	7.45%	.3420	11.76%	.3250	6.21%	.3350	9.48%
400	.1689	8.48%	.1613	3.60%	.1682	8.03%	.3430	12.09%	.3260	6.54%	.3360	9.80%
500	.1692	8.67%	.1632	4.82%	.1699	9.12%	.3420	11.76%	.3260	6.54%	.3360	9.80%
600	.1689	8.48%	.1628	4.56%	.1695	8.86%	.3400	11.11%	.3250	6.21%	.3340	9.15%
700	.1685	8.22%	.1624	4.30%	.1692	8.67%	.3410	11.44%	.3270	6.86%	.3360	9.80%
800	.1692	8.67%	.1633	4.88%	.1701	9.25%	.3420	11.76%	.3270	6.86%	.3370	10.13%
900	.1691	8.61%	.1627	4.50%	.1696	8.93%	.3410	11.44%	.3270	6.86%	.3360	9.80%
1000	.1701	9.25%	.1639	5.27%	.1708	9.70%	.3430	12.09%	.3280	7.19%	.3390	10.78%
Trec8												
N	Avg P(baseline .1964)						P@20(baseline .3640)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2065	5.14%	.2041	3.92%	.2053	4.53%	.3980	9.34%	.3930	7.97%	.3930	7.97%
200	.2097	6.77%	.2061	4.94%	.2073	5.55%	.4000	9.89%	.3860	6.04%	.3870	6.32%
300	.2126	8.25%	.2087	6.26%	.2097	6.77%	.3980	9.34%	.3810	4.67%	.3840	5.49%
400	.2142	9.06%	.2100	6.92%	.2111	7.48%	.3970	9.07%	.3800	4.40%	.3820	4.95%
500	.2146	9.27%	.2101	6.98%	.2110	7.43%	.3970	9.07%	.3830	5.22%	.3840	5.49%
600	.2149	9.42%	.2100	6.92%	.2110	7.43%	.3960	8.79%	.3800	4.40%	.3820	4.95%
700	.2153	9.62%	.2104	7.13%	.2114	7.64%	.3940	8.24%	.3790	4.12%	.3810	4.67%
800	.2170	10.49%	.2120	7.94%	.2130	8.45%	.3940	8.24%	.3770	3.57%	.3790	4.12%
900	.2171	10.54%	.2119	7.89%	.2131	8.50%	.3940	8.24%	.3760	3.30%	.3780	3.85%
1000	.2171	10.54%	.2115	7.69%	.2128	8.35%	.3940	8.24%	.3750	3.02%	.3770	3.57%

Table 20: Reranking with no feedback (Holtz's Method)

Trec6												
N	Avg P(baseline .1966)						P@20(baseline .3180)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2400	22.08%	.2410	22.58%	.2413	22.74%	.3640	14.47%	.3690	16.04%	.3710	16.67%
200	.2395	21.82%	.2372	20.65%	.2389	21.52%	.3660	15.09%	.3710	16.67%	.3720	16.98%
300	.2403	22.23%	.2402	22.18%	.2412	22.69%	.3670	15.41%	.3720	16.98%	.3700	16.35%
400	.2400	22.08%	.2432	23.70%	.2417	22.94%	.3680	15.72%	.3760	18.24%	.3720	16.98%
500	.2403	22.23%	.2469	25.58%	.2424	23.30%	.3630	14.15%	.3860	21.38%	.3750	17.92%
600	.2401	22.13%	.2439	24.06%	.2434	23.80%	.3640	14.47%	.3810	19.81%	.3780	18.87%
700	.2386	21.36%	.2428	23.50%	.2431	23.65%	.3630	14.15%	.3770	18.55%	.3780	18.87%
800	.2386	21.36%	.2428	23.50%	.2431	23.65%	.3630	14.15%	.3730	17.30%	.3770	18.55%
900	.2387	21.41%	.2423	23.25%	.2428	23.50%	.3650	14.78%	.3700	16.35%	.3760	18.24%
1000	.2387	21.41%	.2414	22.79%	.2429	23.55%	.3650	14.78%	.3700	16.35%	.3780	18.87%
Trec7												
N	Avg P(baseline .1557)						P@20(baseline .3060)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2091	34.30%	.2078	33.46%	.2107	35.32%	.3840	25.49%	.3830	25.16%	.3860	26.14%
200	.2118	36.03%	.2061	32.37%	.2089	34.17%	.3850	25.82%	.3790	23.86%	.3800	24.18%
300	.2113	35.71%	.2029	30.31%	.2081	33.65%	.3830	25.16%	.3770	23.20%	.3830	25.16%
400	.2118	36.03%	.2037	30.83%	.2085	33.91%	.3830	25.16%	.3760	22.88%	.3850	25.82%
500	.2137	37.25%	.2046	31.41%	.2107	35.32%	.3850	25.82%	.3780	23.53%	.3800	24.18%
600	.2135	37.12%	.2044	31.28%	.2099	34.81%	.3820	24.84%	.3760	22.88%	.3790	23.86%
700	.2134	37.06%	.2044	31.28%	.2100	34.87%	.3830	25.16%	.3780	23.53%	.3810	24.51%
800	.2135	37.12%	.2041	31.09%	.2101	34.94%	.3820	24.84%	.3760	22.88%	.3800	24.18%
900	.2138	37.32%	.2041	31.09%	.2094	34.49%	.3800	24.18%	.3730	21.90%	.3770	23.20%
1000	.2147	37.89%	.2041	31.09%	.2100	34.87%	.3820	24.84%	.3730	21.90%	.3770	23.20%
Trec8												
N	Avg P(baseline .1964)						P@20(baseline .3640)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2357	20.01%	.2359	20.11%	.2354	19.86%	.4160	14.29%	.4180	14.84%	.4190	15.11%
200	.2343	19.30%	.2332	18.74%	.2334	18.84%	.4170	14.56%	.4180	14.84%	.4200	15.38%
300	.2332	18.74%	.2318	18.02%	.2318	18.02%	.4180	14.84%	.4140	13.74%	.4180	14.84%
400	.2329	18.58%	.2317	17.97%	.2309	17.57%	.4180	14.84%	.4110	12.91%	.4130	13.46%
500	.2327	18.48%	.2321	18.18%	.2306	17.41%	.4200	15.38%	.4120	13.19%	.4180	14.84%
600	.2324	18.33%	.2311	17.67%	.2298	17.01%	.4200	15.38%	.4150	14.01%	.4190	15.11%
700	.2322	18.23%	.2310	17.62%	.2297	16.96%	.4180	14.84%	.4140	13.74%	.4180	14.84%
800	.2324	18.33%	.2312	17.72%	.2297	16.96%	.4180	14.84%	.4130	13.46%	.4170	14.56%
900	.2324	18.33%	.2312	17.72%	.2296	16.90%	.4200	15.38%	.4150	14.01%	.4190	15.11%
1000	.2323	18.28%	.2307	17.46%	.2292	16.70%	.4200	15.38%	.4150	14.01%	.4190	15.11%

Table 21: RRPf; no negative feedback; $\alpha=\beta=8, \gamma=0$ (Holtz's method)

Trec6												
N	Avg P(baseline .1966)						P@20(baseline .3180)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2400	22.08%	.2410	22.58%	.2413	22.74%	.3640	14.47%	.3690	16.04%	.3710	16.67%
200	.2395	21.82%	.2372	20.65%	.2389	21.52%	.3660	15.09%	.3710	16.67%	.3720	16.98%
300	.2403	22.23%	.2402	22.18%	.2412	22.69%	.3670	15.41%	.3720	16.98%	.3700	16.35%
400	.2400	22.08%	.2432	23.70%	.2417	22.94%	.3680	15.72%	.3760	18.24%	.3720	16.98%
500	.2403	22.23%	.2469	25.58%	.2424	23.30%	.3630	14.15%	.3860	21.38%	.3750	17.92%
600	.2496	26.96%	.2567	30.57%	.2582	31.33%	.3770	18.55%	.3910	22.96%	.3880	22.01%
700	.2502	27.26%	.2588	31.64%	.2600	32.25%	.3780	18.87%	.3890	22.33%	.3900	22.64%
800	.2510	27.67%	.2590	31.74%	.2603	32.40%	.3770	18.55%	.3930	23.58%	.3910	22.96%
900	.2505	27.42%	.2584	31.43%	.2599	32.20%	.3760	18.24%	.3940	23.90%	.3930	23.58%
1000	.2504	27.37%	.2584	31.43%	.2600	32.25%	.3760	18.24%	.3940	23.90%	.3930	23.58%
Trec7												
N	Avg P(baseline .1557)						P@20(baseline .3060)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2091	34.30%	.2078	33.46%	.2107	35.32%	.3840	25.49%	.3830	25.16%	.3860	26.14%
200	.2118	36.03%	.2061	32.37%	.2089	34.17%	.3850	25.82%	.3790	23.86%	.3800	24.18%
300	.2113	35.71%	.2029	30.31%	.2081	33.65%	.3830	25.16%	.3770	23.20%	.3830	25.16%
400	.2118	36.03%	.2037	30.83%	.2085	33.91%	.3830	25.16%	.3760	22.88%	.3850	25.82%
500	.2137	37.25%	.2046	31.41%	.2107	35.32%	.3850	25.82%	.3780	23.53%	.3800	24.18%
600	.2331	49.71%	.2263	45.34%	.2306	48.11%	.3860	26.14%	.3810	24.51%	.3800	24.18%
700	.2359	51.51%	.2291	47.14%	.2332	49.78%	.3910	27.78%	.3860	26.14%	.3880	26.80%
800	.2362	51.70%	.2292	47.21%	.2326	49.39%	.3870	26.47%	.3810	24.51%	.3820	24.84%
900	.2360	51.57%	.2292	47.21%	.2329	49.58%	.3850	25.82%	.3810	24.51%	.3850	25.82%
1000	.2374	52.47%	.2298	47.59%	.2339	50.22%	.3900	27.45%	.3840	25.49%	.3870	26.47%
Trec8												
N	Avg P(baseline .1964)						P@20(baseline .3640)					
	CL		SP		IC		CL		SP		IC	
	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B	Value	%B
100	.2357	20.01%	.2359	20.11%	.2354	19.86%	.4160	14.29%	.4180	14.84%	.4190	15.11%
200	.2343	19.30%	.2332	18.74%	.2334	18.84%	.4170	14.56%	.4180	14.84%	.4200	15.38%
300	.2332	18.74%	.2318	18.02%	.2318	18.02%	.4180	14.84%	.4140	13.74%	.4180	14.84%
400	.2329	18.58%	.2317	17.97%	.2309	17.57%	.4180	14.84%	.4110	12.91%	.4130	13.46%
500	.2327	18.48%	.2321	18.18%	.2306	17.41%	.4200	15.38%	.4120	13.19%	.4180	14.84%
600	.2518	28.21%	.2508	27.70%	.2508	27.70%	.4210	15.66%	.4120	13.19%	.4150	14.01%
700	.2528	28.72%	.2520	28.31%	.2521	28.36%	.4180	14.84%	.4140	13.74%	.4190	15.11%
800	.2542	29.43%	.2529	28.77%	.2529	28.77%	.4180	14.84%	.4120	13.19%	.4160	14.29%
900	.2544	29.53%	.2530	28.82%	.2530	28.82%	.4190	15.11%	.4170	14.56%	.4200	15.38%
1000	.2553	29.99%	.2533	28.97%	.2535	29.07%	.4170	14.56%	.4150	14.01%	.4180	14.84%

Table 22: RRPf; $\alpha=\beta=\gamma=8, T=300$; Assumed non-relevant set=rank 501-100

Trec6								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1840	-6.41%	.1754	-10.78%	.3770	18.55%	.3560	11.95%
200	.1840	-6.41%	.1892	-3.76%	.3770	18.55%	.3740	17.61%
300	.1840	-6.41%	.1938	-1.42%	.3770	18.55%	.3710	16.67%
400	.1840	-6.41%	.1942	-1.22%	.3770	18.55%	.3750	17.92%
500	.1840	-6.41%	.1932	-1.73%	.3770	18.55%	.3710	16.67%
600	.1840	-6.41%	.1913	-2.70%	.3770	18.55%	.3680	15.72%
700	.1840	-6.41%	.1912	-2.75%	.3770	18.55%	.3640	14.47%
800	.1840	-6.41%	.1905	-3.10%	.3770	18.55%	.3660	15.09%
900	.1840	-6.41%	.1910	-2.85%	.3770	18.55%	.3670	15.41%
1000	.1840	-6.41%	.1914	-2.64%	.3770	18.55%	.3730	17.30%
Trec7								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1833	17.73%	.1954	25.50%	.3650	19.28%	.3970	29.74%
200	.1833	17.73%	.1982	27.30%	.3650	19.28%	.3890	27.12%
300	.1833	17.73%	.1986	27.55%	.3650	19.28%	.3950	29.08%
400	.1833	17.73%	.1971	26.59%	.3650	19.28%	.3920	28.10%
500	.1833	17.73%	.1962	26.01%	.3650	19.28%	.3890	27.12%
600	.1833	17.73%	.1975	26.85%	.3650	19.28%	.3940	28.76%
700	.1833	17.73%	.1990	27.81%	.3650	19.28%	.4010	31.05%
800	.1833	17.73%	.1985	27.49%	.3650	19.28%	.3980	30.07%
900	.1833	17.73%	.1995	28.13%	.3650	19.28%	.3970	29.74%
1000	.1833	17.73%	.1969	26.46%	.3650	19.28%	.3950	29.08%
Trec8								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1827	-6.98%	.1812	-7.74%	.3820	4.95%	.3710	1.92%
200	.1827	-6.98%	.1789	-8.91%	.3820	4.95%	.3680	1.10%
300	.1827	-6.98%	.1792	-8.76%	.3820	4.95%	.3730	2.47%
400	.1827	-6.98%	.1830	-6.82%	.3820	4.95%	.3760	3.30%
500	.1827	-6.98%	.1820	-7.33%	.3820	4.95%	.3760	3.30%
600	.1827	-6.98%	.1812	-7.74%	.3820	4.95%	.3760	3.30%
700	.1827	-6.98%	.1744	-11.20%	.3820	4.95%	.3650	.27%
800	.1827	-6.98%	.1736	-11.61%	.3820	4.95%	.3650	.27%
900	.1827	-6.98%	.1765	-10.13%	.3820	4.95%	.3610	-.82%
1000	.1827	-6.98%	.1735	-11.66%	.3820	4.95%	.3640	0%

Table 23: Rel. Fdbk. and Hybrid Fdbk; R = 5 and M = 10

Trec6								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1816	-7.63%	.1803	-8.29%	.3710	16.67%	.3470	9.12%
200	.1816	-7.63%	.1837	-6.56%	.3710	16.67%	.3540	11.32%
300	.1816	-7.63%	.1861	-5.34%	.3710	16.67%	.3660	15.09%
400	.1816	-7.63%	.1893	-3.71%	.3710	16.67%	.3600	13.21%
500	.1816	-7.63%	.1884	-4.17%	.3710	16.67%	.3610	13.52%
600	.1816	-7.63%	.1884	-4.17%	.3710	16.67%	.3550	11.64%
700	.1816	-7.63%	.1882	-4.27%	.3710	16.67%	.3540	11.32%
800	.1816	-7.63%	.1871	-4.83%	.3710	16.67%	.3540	11.32%
900	.1816	-7.63%	.1869	-4.93%	.3710	16.67%	.3600	13.21%
1000	.1816	-7.63%	.1870	-4.88%	.3710	16.67%	.3600	13.21%
Trec7								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1893	21.58%	.1846	18.56%	.3650	19.28%	.3750	22.55%
200	.1893	21.58%	.1968	26.40%	.3650	19.28%	.4020	31.37%
300	.1893	21.58%	.1967	26.33%	.3650	19.28%	.4020	31.37%
400	.1893	21.58%	.1964	26.14%	.3650	19.28%	.4030	31.70%
500	.1893	21.58%	.1974	26.78%	.3650	19.28%	.4050	32.35%
600	.1893	21.58%	.1958	25.75%	.3650	19.28%	.4050	32.35%
700	.1893	21.58%	.1957	25.69%	.3650	19.28%	.4060	32.68%
800	.1893	21.58%	.1928	23.83%	.3650	19.28%	.3930	28.43%
900	.1893	21.58%	.1906	22.41%	.3650	19.28%	.3890	27.12%
1000	.1893	21.58%	.1901	22.09%	.3650	19.28%	.3880	26.80%
Trec8								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1740	-11.41%	.1640	-16.50%	.3870	6.32%	.3630	-.27%
200	.1740	-11.41%	.1600	-18.53%	.3870	6.32%	.3690	1.37%
300	.1740	-11.41%	.1592	-18.94%	.3870	6.32%	.3640	0%
400	.1740	-11.41%	.1591	-18.99%	.3870	6.32%	.3700	1.65%
500	.1740	-11.41%	.1594	-18.84%	.3870	6.32%	.3740	2.75%
600	.1740	-11.41%	.1597	-18.69%	.3870	6.32%	.3700	1.65%
700	.1740	-11.41%	.1558	-20.67%	.3870	6.32%	.3670	.82%
800	.1740	-11.41%	.1566	-20.26%	.3870	6.32%	.3690	1.37%
900	.1740	-11.41%	.1581	-19.50%	.3870	6.32%	.3730	2.47%
1000	.1740	-11.41%	.1564	-20.37%	.3870	6.32%	.3720	2.20%

Table 24: Rel. Fdbk. and Hybrid Fdbk; R = 10 and M = 20

Trec6								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1840	-6.41%	.1917	-2.49%	.3770	18.55%	.3670	15.41%
200	.1840	-6.41%	.1991	1.27%	.3770	18.55%	.3740	17.61%
300	.1840	-6.41%	.2069	5.24%	.3770	18.55%	.3840	20.75%
400	.1840	-6.41%	.2045	4.02%	.3770	18.55%	.3800	19.50%
500	.1840	-6.41%	.2036	3.56%	.3770	18.55%	.3780	18.87%
600	.1840	-6.41%	.2001	1.78%	.3770	18.55%	.3720	16.98%
700	.1840	-6.41%	.1999	1.68%	.3770	18.55%	.3710	16.67%
800	.1840	-6.41%	.1986	1.02%	.3770	18.55%	.3700	16.35%
900	.1840	-6.41%	.1987	1.07%	.3770	18.55%	.3700	16.35%
1000	.1840	-6.41%	.1985	.97%	.3770	18.55%	.3700	16.35%
Trec7								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1833	17.73%	.1925	23.64%	.3650	19.28%	.3690	20.59%
200	.1833	17.73%	.2000	28.45%	.3650	19.28%	.3830	25.16%
300	.1833	17.73%	.2018	29.61%	.3650	19.28%	.3800	24.18%
400	.1833	17.73%	.1994	28.07%	.3650	19.28%	.3780	23.53%
500	.1833	17.73%	.2008	28.97%	.3650	19.28%	.3810	24.51%
600	.1833	17.73%	.2005	28.77%	.3650	19.28%	.3800	24.18%
700	.1833	17.73%	.2006	28.84%	.3650	19.28%	.3780	23.53%
800	.1833	17.73%	.1986	27.55%	.3650	19.28%	.3730	21.90%
900	.1833	17.73%	.1981	27.23%	.3650	19.28%	.3720	21.57%
1000	.1833	17.73%	.1955	25.56%	.3650	19.28%	.3680	20.26%
Trec8								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1827	-6.98%	.1780	-9.37%	.3820	4.95%	.3620	-.55%
200	.1827	-6.98%	.1763	-10.23%	.3820	4.95%	.3630	-.27%
300	.1827	-6.98%	.1749	-10.95%	.3820	4.95%	.3630	-.27%
400	.1827	-6.98%	.1740	-11.41%	.3820	4.95%	.3640	0%
500	.1827	-6.98%	.1749	-10.95%	.3820	4.95%	.3620	-.55%
600	.1827	-6.98%	.1760	-10.39%	.3820	4.95%	.3630	-.27%
700	.1827	-6.98%	.1756	-10.59%	.3820	4.95%	.3610	-.82%
800	.1827	-6.98%	.1756	-10.59%	.3820	4.95%	.3620	-.55%
900	.1827	-6.98%	.1756	-10.59%	.3820	4.95%	.3620	-.55%
1000	.1827	-6.98%	.1755	-10.64%	.3820	4.95%	.3610	-.82%

Table 25: Rel. Fdbk. and Hybrid Fdbk; R = 5 and M = 10 (No bad phrases)

Trec6								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1816	-7.63%	.1740	-11.50%	.3710	16.67%	.3270	2.83%
200	.1816	-7.63%	.1779	-9.51%	.3710	16.67%	.3380	6.29%
300	.1816	-7.63%	.1822	-7.32%	.3710	16.67%	.3380	6.29%
400	.1816	-7.63%	.1840	-6.41%	.3710	16.67%	.3420	7.55%
500	.1816	-7.63%	.1846	-6.10%	.3710	16.67%	.3410	7.23%
600	.1816	-7.63%	.1843	-6.26%	.3710	16.67%	.3440	8.18%
700	.1816	-7.63%	.1850	-5.90%	.3710	16.67%	.3470	9.12%
800	.1816	-7.63%	.1857	-5.54%	.3710	16.67%	.3510	10.38%
900	.1816	-7.63%	.1866	-5.09%	.3710	16.67%	.3560	11.95%
1000	.1816	-7.63%	.1866	-5.09%	.3710	16.67%	.3560	11.95%
Trec7								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1893	21.58%	.1690	8.54%	.3650	19.28%	.3360	9.80%
200	.1893	21.58%	.1752	12.52%	.3650	19.28%	.3460	13.07%
300	.1893	21.58%	.1782	14.45%	.3650	19.28%	.3450	12.75%
400	.1893	21.58%	.1788	14.84%	.3650	19.28%	.3570	16.67%
500	.1893	21.58%	.1814	16.51%	.3650	19.28%	.3640	18.95%
600	.1893	21.58%	.1808	16.12%	.3650	19.28%	.3640	18.95%
700	.1893	21.58%	.1809	16.18%	.3650	19.28%	.3680	20.26%
800	.1893	21.58%	.1806	15.99%	.3650	19.28%	.3700	20.92%
900	.1893	21.58%	.1813	16.44%	.3650	19.28%	.3690	20.59%
1000	.1893	21.58%	.1816	16.63%	.3650	19.28%	.3710	21.24%
Trec8								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	Rel. Fdbk		Hybrid Fdbk		Rel. Fdbk		Hybrid Fdbk	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1740	-11.41%	.1571	-20.01%	.3870	6.32%	.3650	.27%
200	.1740	-11.41%	.1579	-19.60%	.3870	6.32%	.3720	2.20%
300	.1740	-11.41%	.1555	-20.82%	.3870	6.32%	.3670	.82%
400	.1740	-11.41%	.1550	-21.08%	.3870	6.32%	.3660	.55%
500	.1740	-11.41%	.1544	-21.38%	.3870	6.32%	.3650	-.27%
600	.1740	-11.41%	.1537	-21.74%	.3870	6.32%	.3610	-.82%
700	.1740	-11.41%	.1535	-21.84%	.3870	6.32%	.3620	-.55%
800	.1740	-11.41%	.1538	-21.69%	.3870	6.32%	.3610	-.82%
900	.1740	-11.41%	.1531	-22.05%	.3870	6.32%	.3600	-1.10%
1000	.1740	-11.41%	.1530	-22.10%	.3870	6.32%	.3640	0%

Table 26: Rel. Fdbk. and Hybrid Fdbk; R = 10 and M = 20(No bad phrases)

Trec6								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2057	4.63%	.2294	16.68%	.3600	13.21%	.3630	14.15%
200	.2110	7.32%	.2258	14.85%	.3640	14.47%	.3600	13.21%
300	.2151	9.41%	.2255	14.70%	.3600	13.21%	.3590	12.89%
400	.2169	10.33%	.2255	14.70%	.3660	15.09%	.3640	14.47%
500	.2177	10.73%	.2267	15.31%	.3660	15.09%	.3670	15.41%
600	.2174	10.58%	.2265	15.21%	.3660	15.09%	.3660	15.09%
700	.2166	10.17%	.2272	15.56%	.3650	14.78%	.3650	14.78%
800	.2169	10.33%	.2278	15.87%	.3660	15.09%	.3670	15.41%
900	.2166	10.17%	.2280	15.97%	.3680	15.72%	.3660	15.09%
1000	.2163	10.02%	.2273	15.62%	.3670	15.41%	.3640	14.47%
Trec7								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1725	10.79%	.2128	36.67%	.3410	11.44%	.3850	25.82%
200	.1755	12.72%	.2184	40.27%	.3480	13.73%	.4040	32.03%
300	.1781	14.39%	.2170	39.37%	.3550	16.01%	.4000	30.72%
400	.1797	15.41%	.2167	39.18%	.3580	16.99%	.3980	30.07%
500	.1814	16.51%	.2177	39.82%	.3610	17.97%	.3940	28.76%
600	.1822	17.02%	.2182	40.14%	.3630	18.63%	.3980	30.07%
700	.1821	16.96%	.2182	40.14%	.3610	17.97%	.4000	30.72%
800	.1832	17.66%	.2192	40.78%	.3640	18.95%	.3980	30.07%
900	.1830	17.53%	.2194	40.91%	.3630	18.63%	.3970	29.74%
1000	.1845	18.50%	.2188	40.53%	.3640	18.95%	.3950	29.08%
Trec8								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2103	7.08%	.2228	13.44%	.3790	4.12%	.3820	4.95%
200	.2115	7.69%	.2186	11.30%	.3720	2.20%	.3830	5.22%
300	.2128	8.35%	.2177	10.85%	.3760	3.30%	.3850	5.77%
400	.2141	9.01%	.2201	12.07%	.3760	3.30%	.3880	6.59%
500	.2137	8.81%	.2190	11.51%	.3740	2.75%	.3870	6.32%
600	.2131	8.50%	.2174	10.69%	.3710	1.92%	.3860	6.04%
700	.2126	8.25%	.2176	10.79%	.3710	1.92%	.3860	6.04%
800	.2142	9.06%	.2180	11.00%	.3730	2.47%	.3860	6.04%
900	.2139	8.91%	.2178	10.90%	.3730	2.47%	.3860	6.04%
1000	.2138	8.86%	.2171	10.54%	.3700	1.65%	.3860	6.04%

Table 27: RR; P2=5,P3=0 and RRPf; no negative feedback; $\alpha=\beta=8,\gamma=0$

Trec6								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2012	2.34%	.2206	12.21%	.3520	10.69%	.3560	11.95%
200	.2077	5.65%	.2132	8.44%	.3530	11.01%	.3400	6.92%
300	.2112	7.43%	.2136	8.65%	.3530	11.01%	.3450	8.49%
400	.2124	8.04%	.2127	8.19%	.3530	11.01%	.3450	8.49%
500	.2124	8.04%	.2117	7.68%	.3520	10.69%	.3440	8.18%
600	.2123	7.99%	.2115	7.58%	.3520	10.69%	.3420	7.55%
700	.2114	7.53%	.2108	7.22%	.3510	10.38%	.3380	6.29%
800	.2116	7.63%	.2114	7.53%	.3530	11.01%	.3390	6.60%
900	.2114	7.53%	.2109	7.27%	.3540	11.32%	.3340	5.03%
1000	.2110	7.32%	.2095	6.56%	.3520	10.69%	.3330	4.72%
Trec7								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1738	11.62%	.2118	36.03%	.3370	10.13%	.3760	22.88%
200	.1760	13.04%	.2179	39.95%	.3470	13.40%	.4020	31.37%
300	.1778	14.19%	.2173	39.56%	.3490	14.05%	.4010	31.05%
400	.1798	15.48%	.2174	39.63%	.3530	15.36%	.3980	30.07%
500	.1811	16.31%	.2183	40.21%	.3530	15.36%	.3940	28.76%
600	.1817	16.70%	.2185	40.33%	.3530	15.36%	.3950	29.08%
700	.1815	16.57%	.2183	40.21%	.3510	14.71%	.3960	29.41%
800	.1825	17.21%	.2191	40.72%	.3530	15.36%	.3960	29.41%
900	.1822	17.02%	.2196	41.04%	.3510	14.71%	.970	29.74%
1000	.1836	17.92%	.2191	40.72%	.3530	15.36%	.3960	29.41%
Trec8								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2087	6.26%	.2203	12.17%	.3710	1.92%	.3830	5.22%
200	.2085	6.16%	.2152	9.57%	.3620	-.55%	.3780	3.85%
300	.2093	6.57%	.2122	8.04%	.3600	-1.10%	.3710	1.92%
400	.2101	6.98%	.2123	8.10%	.3590	-1.37%	.3740	2.75%
500	.2093	6.57%	.2105	7.18%	.3570	-1.92%	.3720	2.20%
600	.2084	6.11%	.2087	6.26%	.3540	-2.75%	.3670	.82%
700	.2073	5.55%	.2086	6.21%	.3540	-2.75%	.3680	1.10%
800	.2087	6.26%	.2086	6.21%	.3530	-3.02%	.3670	.82%
900	.2083	6.06%	.2081	5.96%	.3500	-3.85%	.3680	1.10%
1000	.2080	5.91%	.2079	5.86%	.3490	-4.12%	.3670	.82%

Table 28: RR; P2=5,P3=5 and RRPF; no negative feedback; $\alpha=\beta=8,\gamma=0$

Trec6								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2105	7.07%	.2400	22.08%	.3510	10.38%	.3640	14.47%
200	.2134	8.55%	.2395	21.82%	.3460	8.81%	.3660	15.09%
300	.2200	11.90%	.2403	22.23%	.3440	8.18%	.3670	15.41%
400	.2194	11.60%	.2400	22.08%	.3440	8.18%	.3680	15.72%
500	.2201	11.95%	.2403	22.23%	.3430	7.86%	.3630	14.15%
600	.2198	11.80%	.2401	21.13%	.3410	7.23%	.3640	14.47%
700	.2193	11.55%	.2386	21.36%	.3400	6.92%	.3630	14.15%
800	.2191	11.44%	.2386	21.36%	.3400	6.92%	.3630	14.15%
900	.2188	11.29%	.2387	21.41%	.3380	6.29%	.3650	14.78%
1000	.2186	11.19%	.2387	21.41%	.3380	6.29%	.3650	14.78%
Trec7								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1662	6.74%	.2091	34.30%	.3410	11.44%	.3840	25.49%
200	.1677	7.71%	.2118	36.03%	.3370	10.13%	.3850	25.82%
300	.1690	8.54%	.2113	35.71%	.3420	11.76%	.3830	25.16%
400	.1689	8.48%	.2118	36.03%	.3430	12.09%	.3830	25.16%
500	.1692	8.67%	.2137	37.25%	.3420	11.76%	.3850	25.82%
600	.1689	8.48%	.2135	37.12%	.3400	11.11%	.3820	24.84%
700	.1685	8.22%	.2134	37.06%	.3410	11.44%	.3830	25.16%
800	.1692	8.67%	.2135	37.12%	.3420	11.76%	.3820	24.84%
900	.1691	8.61%	.2138	37.32%	.3410	11.44%	.3800	24.18%
1000	.1701	9.25%	.2147	37.89%	.3430	12.09%	.3820	24.84%
Trec8								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2089	6.36%	.2160	9.98%	.3690	1.37%	.3720	2.20%
200	.2081	5.96%	.2112	7.54%	.3560	-2.20%	.3630	.27%
300	.2089	6.36%	.2098	6.82%	.3550	-2.47%	.3710	1.92%
400	.2106	7.23%	.2104	7.13%	.3580	-1.65%	.3680	1.10%
500	.2095	6.67%	.2090	6.42%	.3570	-1.92%	.3660	.55%
600	.2082	6.01%	.2073	5.55%	.3500	-3.85%	.3610	.82%
700	.2070	5.40%	.2069	5.35%	.3500	-3.85%	.3590	1.37%
800	.2081	5.96%	.2066	5.19%	.3490	-4.12%	.3620	.55%
900	.2077	5.75%	.2062	4.99%	.3450	-5.22%	.3610	.82%
1000	.2071	5.45%	.2061	4.94%	.3450	-5.22%	.3600	1.10%

Table 29: RR; P2=10, P3=5 and RRPF; no negative feedback; $\alpha=\beta=8, \gamma=0$

Trec6 (P2 = 10, P3 = 5)								
N	Avg P(baseline .1966)				P@20(baseline .3180)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2105	7.07%	.2400	22.08%	.3510	10.38%	.3640	14.47%
200	.2134	8.55%	.2395	21.82%	.3460	8.81%	.3660	15.09%
300	.2200	11.90%	.2403	22.23%	.3440	8.18%	.3670	15.41%
400	.2194	11.60%	.2400	22.08%	.3440	8.18%	.3680	15.72%
500	.2201	11.95%	.2403	22.23%	.3430	7.86%	.3630	14.15%
600	.2198	11.80%	.2496	26.96%	.3410	7.23%	.3770	18.55%
700	.2193	11.55%	.2502	27.26%	.3400	6.92%	.3780	18.87%
800	.2191	11.44%	.2510	27.67%	.3400	6.92%	.3770	18.55%
900	.2188	11.29%	.2505	27.42%	.3380	6.29%	.3760	18.24%
1000	.2186	11.19%	.2504	27.37%	.3380	6.29%	.3760	18.24%
Trec7 (P2 = 5, P3 = 0)								
N	Avg P(baseline .1557)				P@20(baseline .3060)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.1725	10.79%	.2128	36.67%	.3410	11.44%	.3850	25.82%
200	.1755	12.72%	.2184	40.27%	.3480	13.73%	.4040	32.03%
300	.1781	14.39%	.2170	39.37%	.3550	16.01%	.4000	30.72%
400	.1797	15.41%	.2167	39.18%	.3580	16.99%	.3980	30.07%
500	.1814	16.51%	.2177	39.82%	.3610	17.97%	.3940	28.76%
600	.1822	17.02%	.2354	51.19%	.3630	18.63%	.3980	30.07%
700	.1821	16.96%	.2372	52.34%	.3610	17.97%	.3970	29.74%
800	.1832	17.66%	.2386	53.24%	.3640	18.95%	.4010	31.05%
900	.1830	17.53%	.2397	53.95%	.3630	18.63%	.3990	30.39%
1000	.1845	18.50%	.2389	53.44%	.3640	18.95%	.3940	28.76%
Trec8 (P2 = 5, P3 = 0)								
N	Avg P(baseline .1964)				P@20(baseline .3640)			
	RR only		RRPF		RR only		RRPF	
	Value	%B	Value	%B	Value	%B	Value	%B
100	.2103	7.08%	.2228	13.44%	.3790	4.12%	.3820	4.95%
200	.2115	7.69%	.2186	11.30%	.3720	2.20%	.3830	5.22%
300	.2128	8.35%	.2177	10.85%	.3760	3.30%	.3850	5.77%
400	.2141	9.01%	.2201	12.07%	.3760	3.30%	.3880	6.59%
500	.2137	8.81%	.2190	11.51%	.3740	2.75%	.3870	6.32%
600	.2131	8.50%	.2359	20.11%	.3710	1.92%	.3880	6.59%
700	.2126	8.25%	.2384	21.38%	.3710	1.92%	.3910	7.42%
800	.2142	9.06%	.2398	22.10%	.3730	2.47%	.3910	7.42%
900	.2139	8.91%	.2401	22.25%	.3730	2.47%	.3890	6.87%
1000	.2138	8.86%	.2403	22.35%	.3700	1.65%	.3900	7.14%

Table 30: RR;RRPF; $\alpha=\beta=8, \gamma=8$

Trec6									
Bin	#Q	PF($\alpha=\beta=8, \gamma=0$)		Rel.Fdbk (R=20)		Hybrid Fdbk(R=5,M=10)			
						N=400 (With Bad Phrases)		N=300 (Without Bad Phrases)	
		Helped	Hurt	Helped	Hurt	Helped	Hurt	Helped	Hurt
0	7	0	0	1	0	1	0	2	0
1	4	0	0	1	2	0	1	0	1
2	7	3	1	6	1	3	2	4	2
3	5	3	2	4	1	3	2	3	2
4	1	0	0	0	1	1	0	0	1
5	5	2	1	3	2	3	2	3	2
6	1	0	1	0	1	0	1	0	1
7	2	2	0	2	0	1	1	1	0
8	2	0	2	0	1	1	1	1	1
9	0	0	0	0	0	0	0	0	0
10	2	1	1	1	0	2	0	2	0
11	3	1	2	0	3	0	3	0	2
12	1	0	1	0	1	0	1	0	1
13	2	2	0	1	1	1	1	1	1
14	3	2	1	1	2	2	1	2	1
15	1	1	0	0	0	1	0	1	0
16	1	1	0	1	0	1	0	1	0
17	1	1	0	0	1	1	0	0	0
18	0	0	0	0	0	0	0	0	0
19	1	0	0	1	0	1	0	1	0
20	1	0	0	0	1	0	0	0	0
Total	50	19	12	22	18	22	16	22	15

Table 31: Trec6; PF, RF, Hybrid Feedback; Queries helped and hurt based on P@20

Trec6									
Bin	#Q	PF ($\alpha=\beta=8, \gamma=0$)		Rel.Fdbk (R=20)		N=800, P2=5, P3=0			
						RR(SP)		RRPF ($\alpha=\beta=8, \gamma=0$)	
		Helped	Hurt	Helped	Hurt	Helped	Hurt	Helped	Hurt
0	7	0	0	1	0	3	0	3	0
1	4	0	0	1	2	1	1	0	2
2	7	3	1	6	1	4	2	4	1
3	5	3	2	4	1	3	0	3	1
4	1	0	0	0	1	1	0	0	0
5	5	2	1	3	2	2	1	2	2
6	1	0	1	0	1	0	0	0	1
7	2	2	0	2	0	0	1	1	1
8	2	0	2	0	1	1	1	0	2
9	0	0	0	0	0	0	0	0	0
10	2	1	1	1	0	1	0	2	0
11	3	1	2	0	3	0	2	0	3
12	1	0	1	0	1	0	0	0	0
13	2	2	0	1	1	1	0	2	0
14	3	2	1	1	2	2	0	2	1
15	1	1	0	0	0	1	0	1	0
16	1	1	0	1	0	1	0	1	0
17	1	1	0	0	1	1	0	1	0
18	0	0	0	0	0	0	0	0	0
19	1	0	0	1	0	1	0	1	0
20	1	0	0	0	1	0	0	0	0
Total	50	19	12	22	18	23	8	23	14

Table 32: Trec6; PF, RF, RR and RRPf; Queries helped and hurt based on P@20

Trec7									
Bin	#Q	PF($\alpha=\beta=8, \gamma=0$)		Rel.Fdbk (R=20)		Hybrid Fdbk(R=5,M=10)			
						N=400 (With Bad Phrases)		N=300 (Without Bad Phrases)	
		Helped	Hurt	Helped	Hurt	Helped	Hurt	Helped	Hurt
0	7	1	0	2	0	4	0	5	0
1	4	2	0	4	0	3	0	2	1
2	4	1	1	3	0	2	1	2	1
3	5	1	2	4	1	2	3	2	3
4	4	2	0	4	0	4	0	3	1
5	2	1	1	1	1	0	1	1	1
6	4	2	0	1	3	2	1	2	1
7	1	0	1	1	0	1	0	1	0
8	2	2	0	1	0	0	0	1	1
9	3	3	0	1	2	0	3	1	2
10	4	2	2	2	2	2	2	3	1
11	2	1	1	1	0	2	0	1	1
12	1	1	0	0	1	1	0	1	0
13	2	2	0	2	0	2	0	2	0
14	1	1	0	0	1	1	0	1	0
15	1	1	0	1	0	1	0	1	0
16	2	1	0	0	2	0	1	0	1
17	0	0	0	0	0	0	0	0	0
18	1	1	0	0	1	0	1	0	1
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0
Total	50	25	8	28	14	27	13	29	15

Table 33: Trec7; PF, RF, Hybrid Feedback; Queries helped and hurt based on P@20

Trec7									
Bin	#Q	PF ($\alpha=\beta=8, \gamma=0$)		Rel.Fdbk (R=20)		N=800, P2=5, P3=0			
						RR(SP)		RRPF ($\alpha=\beta=8, \gamma=0$)	
		Helped	Hurt	Helped	Hurt	Helped	Hurt	Helped	Hurt
0	7	1	0	2	0	3	0	3	0
1	4	2	0	4	0	3	0	3	0
2	4	1	1	3	0	3	0	2	0
3	5	1	2	4	1	3	2	2	2
4	4	2	0	4	0	2	0	2	1
5	2	1	1	1	1	2	0	1	1
6	4	2	0	1	3	2	2	2	1
7	1	0	1	1	0	1	0	1	0
8	2	2	0	1	0	0	1	2	0
9	3	3	0	1	2	2	0	2	1
10	4	2	2	2	2	3	1	3	1
11	2	1	1	1	0	0	1	2	0
12	1	1	0	0	1	0	0	1	0
13	2	2	0	2	0	1	0	2	0
14	1	1	0	0	1	0	1	1	0
15	1	1	0	1	0	1	0	1	0
16	2	1	0	0	2	2	0	2	0
17	0	0	0	0	0	0	0	0	0
18	1	1	0	0	1	0	1	0	1
19	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0
Total	50	25	8	28	14	28	9	32	8

Table 34: Trec7; PF, RF, RR and RRPf; Queries helped and hurt based on P@20

Trec8									
Bin	#Q	PF($\alpha=\beta=8, \gamma=0$)		Rel.Fdbk (R=20)		Hybrid Fdbk(R=5,M=10)			
						N=400 (With Bad Phrases)		N=300 (Without Bad Phrases)	
		Helped	Hurt	Helped	Hurt	Helped	Hurt	Helped	Hurt
0	4	1	0	3	0	2	0	3	0
1	3	1	1	1	1	1	1	1	1
2	3	1	1	2	1	1	2	1	1
3	3	1	1	2	1	2	1	1	1
4	3	2	1	2	0	3	0	3	0
5	7	5	1	5	2	4	1	4	1
6	3	2	0	1	2	1	1	2	1
7	4	3	1	3	1	2	2	2	2
8	2	1	1	1	1	1	1	1	1
9	3	2	1	0	3	1	2	1	2
10	3	1	0	0	3	0	3	0	3
11	3	2	1	2	1	2	1	2	1
12	0	0	0	0	0	0	0	0	0
13	1	1	0	1	0	1	0	1	0
14	1	0	0	0	1	0	1	0	1
15	2	2	0	1	1	0	2	0	2
16	1	0	0	1	0	0	1	0	1
17	1	1	0	0	1	0	0	0	0
18	1	0	0	0	1	0	1	0	1
19	1	0	1	0	1	0	1	0	1
20	1	0	0	0	1	0	0	0	0
Total	50	26	10	25	22	21	21	22	20

Table 35: Trec8; PF, RF, Hybrid Feedback; Queries helped and hurt based on P@20

Trec8									
Bin	#Q	PF ($\alpha=\beta=8, \gamma=0$)		Rel.Fdbk (R=20)		N=800, P2=5, P3=0			
						RR(SP)		RRPF ($\alpha=\beta=8, \gamma=0$)	
		Helped	Hurt	Helped	Hurt	Helped	Hurt	Helped	Hurt
0	4	1	0	3	0	2	0	2	0
1	3	1	1	1	1	2	0	1	2
2	3	1	1	2	1	0	1	1	2
3	3	1	1	2	1	1	2	1	1
4	3	2	1	2	0	1	2	2	1
5	7	5	1	5	2	4	1	5	1
6	3	2	0	1	2	2	0	3	0
7	4	3	1	3	1	3	1	3	1
8	2	1	1	1	1	0	2	1	1
9	3	2	1	0	3	3	0	3	0
10	3	1	0	0	3	0	0	1	2
11	3	2	1	2	1	1	2	1	2
12	0	0	0	0	0	0	0	0	0
13	1	1	0	1	0	0	1	0	1
14	1	0	0	0	1	0	0	0	0
15	2	2	0	1	1	0	2	0	1
16	1	0	0	1	0	0	1	0	1
17	1	1	0	0	1	1	0	1	0
18	1	0	0	0	1	0	1	0	0
19	1	0	1	0	1	0	1	0	1
20	1	0	0	0	1	0	1	0	0
Total	50	26	10	25	22	20	18	25	17

Table 36: Trec8; PF, RF, RR and RRPf; Queries helped and hurt based on P@20