TREC-7 Ad-Hoc, High Precision and Filtering Experiments using PIRCS

K.L. Kwok, L. Grunfeld, M. Chan & N. Dinstl

Computer Science Department
Queens College, CUNY

C. Cool

Graduate School of Library & Information Studies
Queens College, CUNY

Abstract

In TREC-7, we participated in the main task of automatic ad-hoc retrieval as well as the high precision and filtering tracks. For ad-hoc, three experiments were done with query types of short (title section of a topic), medium (description section) and long (all sections) lengths. We used a sequence of five methods to handle the short and medium length queries. For long queries we employed a re-ranking method based on evidence from matching query phrases in document windows in both stages of a 2-stage retrieval. Results are well above median. For high precision track, we employed our interactive PIRCS system for the first time. In adaptive filtering, we concentrate on dynamically varying the retrieval status value threshold for deciding selection and during the course of filtering. Query weights were trained but expansion was not done. We also submitted results for batch filtering and standard routing based on methods evolved from previous TREC experiments.

1. Introduction

We continue to use our PIRCS system for experimentation. This system has been described in previous TREC proceedings and summarized in TREC-6. For TREC-7, we participated in the main task of automatic ad-hoc retrieval as well as the high precision and filtering tracks. Two strategies we used throughout for ad-hoc are 2-stage retrieval and collection enrichment. 2-stage retrieval means using a raw query for initial retrieval, employ the top-ranked document list to define the domain of the query and expand the initial query in a pseudo-relevance feedback fashion, and then do a 2nd retrieval to report final results. Collection enrichment means adding external documents to the target collection during 1st stage in order to enhance the quality of the pseudo-feedback documents. Both strategies have been found to work more often than not for queries of different lengths and applicable to different languages such as English and Chinese. Ad-hoc retrievals are discussed in Section 2.

This year we introduce an interactive PIRCS system, and users can interrogate the TREC-7 collection via a GUI that is based on the look and feel of ZPRISE from NIST. We used this facility for our High Precision track to find as many relevant documents as possible within the top fifteen retrieved. Results are reported in Section 3.

We also participated in the Filtering track, in particular, adaptive filtering being done the first time. For this, we emphasize on dynamically setting a RSV (retrieval status value) threshold to select or not select a document for examination. Adaptive filtering is described in Section 4 while batch filtering and routing work are described in Section 5. Section 6 contains our conclusions.

2 Ad-Hoc Retrieval

The target collection for ad hoc retrieval is Disks 4&5, similar to TREC-6 except that Congressional Record documents are not considered. This leaves the collection to consist of Financial Times, Federal Register, Foreign Broadcast Information Service and the LA Times, some 1.9 GB of text in over ½ million documents. We did not bother to re-create a new dictionary and continue to use last year's data that include terms and statistics from the Congressional Record. The documents however were removed from retrieval. As usual, we broke long documents into sub-documents of about 550 words long ending on a paragraph boundary, resulting in some 761K sub-documents. The dictionary of unique terms is 727K in size after stopword removal and conflation; it became 325K in size after Zipf thresholding.

As in previous TRECs, topics for TREC-7 are described in several sections: title, description and narrative. This year, each section contains self-sufficient information so that queries can be constructed from only the Title (short length), the Description (medium) or from All sections (long). Their average number of unique terms after stopword removal, stem conflation and thresholding are 2.62, 7.04 and 17.18 respectively. Short queries lack descriptive term variety and term weighting information and pose especially difficult situations for a search engine. Queries using All sections are paragraph(s) in size. One does not expect casual users to issue such long queries. On the other hand analysts, information specialists or dedicated information seekers should not refrain from elaborating their needs in such verbose fashion with variety and redundancy, since long queries usually give better results than short. In this experiment, we treat short and medium queries similarly and are discussed in Section 2.1. Long queries are described in Section 2.2.

2.1 Short and Medium Queries

Over the past several years our efforts to investigate short query retrievals have accumulated in [KwCh98] where we applied five methods in sequence to bring effectiveness some 32 to 72% above the initial 1-stage retrieval results for TREC-6&5 environments. These same procedures were used in TREC-7, and it provides a blind experimental test of the approach. The methods we employed are: 1) avtf query term weighting, 2) variable high frequency Zipfian threshold, 3) collection enrichment, 4) enhancing term variety in raw queries, and 5) using retrieved document local term statistics. Avtf employs collection statistics to weight terms in short queries [Kwok96]. Variable high frequency threshold defines and ignores statistical stopwords based on query length. Collection enrichment adds other collections to the one under investigation so as to improve the chance of ranking more relevant documents in the top n for the pseudo-feedback process [WRBJJ98, KwCh98]. Enhancing term variety to raw queries means finding highly associated terms in the initially retrieved documents that are domain-related to the query [XuCr96]. Making the query longer may improve another round of 1st stage retrieval. And retrieved document local statistics re-weight terms in the 2nd stage using the set of domain-related documents rather than the whole collection as used during the initial stage [SiMB97]. For collection enrichment, we form a miscellaneous collection by retrieving the top 200 documents from the sub-collections of Disks 1-3 using the short form of the queries. This miscellaneous collection is used to enrich the retrieval operation during the initial stage retrieval. Some of these documents are ranked high and they are employed to expand the queries during pseudo-relevance feedback.

Query Type

	Title % Short		Desc % Medium		All % Long	
Relv.Ret (at most)	2983 (4674)	0	3034 (4674)	2	3162 (4674)	6
Avg.Prec	.2427	0	.2543	5	.2723	12
P@10	.4480	0	.4600	3	.4960	11
P@20	.3770	0	.3930	4	.4340	15
P@30	.3353	0	.3613	8	.3947	18
R.Prec	.2705	0	.2831	5	.2960	9

Table 1: Automatic Ad Hoc Results for 50 Queries

Results and Discussion

Our TREC-7 results for short and medium queries are summarized in Table 1 under the columns Title and Desc. These runs are named 'pirc8At' and 'pirc8Ad' respectively. This year, all important terms that appear in the Title section are also repeated in the Description section. The official evaluation measure is average non-interpolated precision (Avg.Prec) which are 0.2427 and 0.2543 respectively. The number of relevants recovered at 1000 documents (Relv.Ret) is respectively 2983 and 3034 out of a maximum of 4674, or about 64% and 65%. The behavior of the two retrievals are very close to each other, with the Desc queries giving an edge of between 2 to 8% uniformly in various average precision values. The number of medium length queries performing better than or about equal to short ones in average non-interpolated precision is 24 out of 50. However, for relevants retrieved at 1000, this number is 40 out of 50. Thus, most medium length queries have larger recall.

Comparisons with the all-sites median average precision, precision at 100 and 1000 documents are given in Table 2. The average precision for short queries is better or equal to median in 33 instances with 2 queries achieving the best, and are worse than median in 17 instances. Similar behavior is also observed for medium length queries. For example, the precision at 100 documents for medium length queries is better or equal to median in 39 instances with 4 being the best, and is worse than median in 11 instances with 1 case being the worst. The median was calculated using results from all automatic ad-hoc experiments without differentiation of query lengths. Thus, our results are well above median. This year there are 7 queries with a single term. Two of these (#364 "rabies" and #392 "robotics")

Query Type

	Title Short	Desc Medium	All Long
	> = <	> = <	> = <
Avg.Prec	32,2 1 17	33,3 0 17	42,3 0 8
RR@100	27,1 8 15,2	30,4 9 11,1	36,2 7 7,2
RR@1K	37,7 2 11	38,13 4 8	44,10 2 4

Table 2: Ad-Hoc Results: Comparison with Median

	← Zipf Threshold 65K	→		← Zipf Variable Threshold, max 100K				→
Method	**0**		**1**	**2**	**3**	**4**	**5a**	**5b**
	1st Stage	2nd Stage	2nd	2nd	2nd	2nd	2nd	2nd
				TREC-7	**Short**			
Relv.Retr	2299	**2964**	2960	2989	2971	2984	**2983**	2965
Avg.Prec	.180	**.226**	.228	.227	.236	.242	**.243**	.238
P@10	.386	**.408**	.406	.398	.434	.446	**.448**	.430
				TREC-6	**Short**			
Relv.Retr	2188	**2272**	2384	2517	2656	2738	**2739**	2691
Avg.Prec	.220	**.240**	.258	.258	.284	.289	**.291**	.290
P@10	.334	**.372**	.402	.388	.444	.442	**.450**	.440
				TREC-7	**Medium**			
Relv.Retr	2285	**2982**	3075	3067	2987	3062	**3034**	2978
Avg.Prec	.167	**.240**	.236	.231	.238	.253	**.254**	.240
P@10	.370	**.454**	.470	.452	.458	.462	**.460**	.452
				TREC-5	**Medium**			
Relv.Retr	1763	**2279**	2335	2635	2732	2787	**2792**	2735
Avg.Prec	.140	**.161**	.181	.214	.234	.239	**.241**	.234
P@10	.290	**.284**	.326	.372	.382	.404	**.406**	.378

Method 0: Standard 1st & 2nd Stage Retrieval; 1: add avtf weighting; 2: add variable Zipf threshold; 3: add collection enrichment; 4: add term variety; 5a: add local statistics weighting; 5b: 5a without adding term variety.

Table 3: Comparing TREC-7 Short & Medium Length Query Results with TREC-6 and TREC-5

have average precision 0.418 and 0.375 respectively. The rest have values below 0.2. This is unlike TREC-6 where there were very specific single term queries (like #312 "hydroponics", #316 "polygamy" and #348 "agoraphobia") that achieve average precision well above 0.5 in our system. Thus, in year to year average precision comparison, one should take this into account.

Table 3 tabulates retrieval results as each of the five methods discussed earlier is sequentially applied. This is compared with similar results achieved for TREC-6&5 queries and reported in [KwCh98]. It is observed that both Methods 1 (avtf weighting) and 3 (Collection enrichment) produce positive but less pronounced effects for TREC-7 than for TREC-6. Otherwise the behavior is similar and we do get about 8% improvements over standard 2-stage retrieval that does not use any of the 5 methods (i.e. Method 5a vs. Method 0 average precision of 0.243 vs. 0.226).

Comparison of the medium length query results with TREC-5 data is less favorable. Both the avtf weighting (Method 1) and the variable Zipf threshold (Method 2) precision decreases compared with not using them (Method 0). It is possible that parameters set for short queries of very few terms should not be the same as for medium queries. However, the collection enrichment (Method 3) and adding term variety (Method 4) improve the result. The overall improvement is about 6% (i.e. Method 5a vs. Method 0 average precision of 0.254 vs. 0.240).

2.2 Long Queries

For long queries, we use all sections of each topic and the run is named 'pirc8Aa2'. The average number of unique terms after the usual processing is 17.18. Of the five short-query techniques described previously, we retain two, viz. collection enrichment and local statistics for irrelevant documents. The high Zipf threshold is fixed at 100K. On the other hand we added a phrase re-ranking methodology. The assumption is that query phrases represent more precise concepts than single terms. If these phrases are found within small document windows, this might contribute additional evidence of relevance and may help re-rank more relevant documents to the top of the retrieval list. This procedure can be employed at the initial retrieval stage to provide more effective pseudo-relevance feedback to improve 2nd stage retrieval. It can also be employed for the final retrieval list to report better results.

Employing phrases to improve retrieval has a long history since the early days of the SMART project. With the recent Tipster and TREC programs, large corpora become available, and there are ambitious attempts to use natural language techniques to discover more precise phrases from gigabyte-size collections which can take days of processing time [e.g. STCM95]. However, it appears that the promise of better retrieval by phrases has not yet materialized as witnessed by years of TREC blind retrieval experiments

	← Standard	→		←	Phrase Re-Rank	→
	1st Stage A	2nd Stage B	Coll Enrich 2nd Stage C	Rerank C C'	Rerank 1st, 2nd Stage D	Rerank D D'
			TREC-7	**Long**		
Relv.Retr.	2717	**3142**	**3165**	3167	3160	**3162**
Avg.Prec	.214	**.257**	**.266**	.269	.271	**.272**
P@10	.466	**.486**	**.502**	.488	.518	**.496**
			TREC-6	**Long**		
Relv.Retr.	2537	**2947**	**3043**	3064	3074	**3088**
Avg.Prec	.237	**.264**	**.305**	.310	.304	**.308**
P@10	.402	**.452**	**.492**	.498	.488	**.490**
			TREC-5	**Long**		
Relv.Retr.	2463	**3077**	**3034**	3049	3052	**3072**
Avg.Prec	.220	**.253**	**.262**	.265	.270	**.273**
P@10	.404	**.414**	**.438**	.440	.446	**.444**

Table 4: Effect of Collection Enrichment & Phrase Re-Ranking on Long Query Retrieval

where the best retrievals are mainly term based.

With this background, we also tried our hands on using phrases but with a less ambitious goal. Our opinion is that weighted term level retrieval is simple and has provided most rewarding results and should form the basis of any retrieval. Phrases may be used to provide 2nd order effects of improvements based on re-ranking of retrieval lists. The difficulty is how to weight their contribution with respect to term level effects so as to achieve cooperation between the two levels of evidence. This is explored empirically by performing a number of experiments.

Given a topic, we first used our standard PIRCS processing to perform a 2-stage retrieval with collection enrichment and local statistics weighting in the final stage. The same topic is then processed by the Mitre POS-tagger to tag each word syntactically, and a simple program is used to bracket noun phrases. We kept only contiguous adjectives and nouns of two or more words long as these presumably would be most content bearing. These phrases are then searched in each document of a retrieval list to discover their occurrence pattern. Only query phrases that occur within a 3-sentence window of a document are considered; any occurrences of longer spans are ignored.

Re-ranking is based on adding a factor that is proportional to the original RSV of the document in a retrieval list, thus:

RSV = RSV * (1 + phr-factor)

phr-factor = constant * Matching * Coverage

where Matching = M(.) is a function that evaluates how good the query phrases are matched within document windows and sum their contributions. When there are matches of three or more terms of a phrase, it is worth twice as much as two-term matches. In addition, the number of times such matching occurs and the size of the query phrase also influence the evaluation. Coverage = C(.) is another function that evaluates how complete a document covers the entire query concept phrases. Below certain thresholds, no re-ranking takes place. For example, if there are two or more three-term matching phrases, the threshold is satisfied. If there are only two-term matching phrases, then the threshold is graded according to the total number of phrases in a query.

Results and Discussion

Long query results are also shown in Tables 1 and 2. The average precision value is 0.2723, and the relevants retrieved at 1000 documents is 3162, about 68% of the maximum 4674. When compared with the all-sites median, our average precision is better in 42 instances with 3 cases being best, and 8 instances are below median. This appears also to be quite good achievement.

Table 4 shows the successive effects of collection enrichment (Column C) and phrase re-ranking on retrieval (D') for these long queries. They add 3.5% (0.266 vs. 0.257) and 2% (0.272 vs. 0.266) respectively to the average precision measure. The behavior is quite similar to what is found in TREC-6 & 5 which are also shown. Column C' and D depicts the intermediate steps of phrase re-ranking: C' shows the effect of re-ranking 2nd stage retrieval only, while D shows re-ranking 1st stage without doing re-ranking on 2nd stage. It seems preferable to do both.

Comparing the results in Table 1 and 2, it can be seen that for this set of queries and documents, long queries are uniformly preferable in average precision to short (between 6% to 18% better), and medium queries performed somewhere in-between (2% to 8% better than short) using our PIRCS system. Compared to short queries, 34 out of 50 instances have better or equal average non-interpolated precision. This becomes 33 out of 50 when compared to median. However, when the number of relevants retrieved at 1000 is used as the measure for comparison, then ALL long queries are better or equal to short and medium queries. It appears that to obtain good recall, long query length is an especially important factor: the longer the better.

The average precision varies between 24 to 27 percent from short to long queries. This level of effectiveness is far from ideal and needs to be improved by further research and investigation. On the other hand, the precision at 10 documents varies between .448 to .496, meaning that between 4 to 5 documents out of the top 10 retrieved are relevant. For many users who do not need exhaustive retrieval, this represents reasonable performance.

3. High Precision Track

Description of the experimental condition: a single volunteer searched all 50 ad-hoc queries. The searcher in this experiment was a female who had five years of experience using online information retrieval systems. The searcher held a Ph.D. degree and was an assistant professor in the Graduate School of Library Studies at Queens College. At the time of the experiment, the ad-hoc queries were new to the searcher. No prior searching had been performed on any of the topics. The searcher had no prior experience searching on the PIRCS system. The task given to the searcher was to "create a final query that will produce a retrieved set in which the top 15 documents has the highest precision." The single searcher in this experiment employed the following strategy in order to maximize high precision results. First, each query and narrative were read, and a small number of topically relevant words were entered in the query window. Top ranked documents in the retrieved set were quickly examined for relevance, and the full text then displayed in order to identify additional terms for use in query expansion, as well as terms to eliminate. The large size of the full text display window, combined with highlighting of the query terms, facilitated quick examination of the performance of each query, in its particular context within the text. Automatic query expansion was not supported at the time by the retrieval system, so query reformulation was done manually. After every query iteration, the ranked list of documents was judged by the searcher for relevance. Non-relevant documents were marked, and subsequently removed from the ranked list. In other words, the searcher did not "save" individual relevant documents along the way over the course of the search episode. Instead, the searcher created one final query that would retrieve the best possible ranked list, after eliminating the marked non-relevant documents.

Searching time: Each of the 50 queries was searched within the five-minute time limit established by the Track. The number of iterations for each topic search ranged from one to five, with an overall average of 2.1.

Results and Discussion

Precision at 15 documents ranged from .00 to 1.00, with an average overall precision of .48. In a comparison of search results across all seven systems in the high precision track, Table.5, our system performed at or above the median in 48% of the queries and below the median in 52% of queries. Topics # 365 and 372 were the easiest queries for the searcher to search, and attained the highest performance levels, with 100% precision at top 15. No relevant documents were retrieved for Topic 353, due to a misspelling of the query term "Antarctica."

	Above	**At median**	**Below**
Precision @ 15	**8**	**16**	**26**

Table 5: High Precision - Comparison with Median

4. Adaptive Filtering Track

We participated in all three sub-tracks within this year's filtering track, viz.: adaptive filtering, routing and batch filtering. They made use of TREC-1 queries 1 to 50 and run against the target AP88, 89 and 90 collections. This section describes our effort in adaptive filtering. Routing and batch filtering are described in Section 5.

We actually did filtering via sub-documents since our old files have already segmented documents. If any or multiple sub-documents of the same document are selected, we regard it as just one document selected. RSV's of different sub-documents are not added, as is done for ad-hoc.

Many considerations are needed for adaptive filtering. These include defining an initial query together with an initial filtering threshold to start the process, adaptively train the query to tailor to the relevant document types seen so far, dynamically change the threshold to select or not select a document for examination, determine how often these changes are to be performed, and at the same time attempt to maximize a target utility value. The task is made more difficult by excluding any use of the topics or the target collections for initial training purposes. The overall properties of the terms encountered (such as collection term frequency, document frequency of terms, etc.) can be captured after the documents have gone through the filtering process, but relevancy information can be used only if a document has been selected for examination. These data can subsequently be used for training purposes.

To prepare for filtering, a dictionary was defined by processing some 1.2 GB of texts consisting of LA, WSJ-1, DOE and ZIFF-1 collections. These were chosen to reflect the 1988-1990 period news articles as well as introducing some scientific and computer terminology into the dictionary with reasonable term usage statistics. The dictionary size after stopword removal, conflation and Zipf thresholding is about 240K. To help debug some new programs, a mirror problem was set up using queries numbered 51 to 100 for filtering a target collection consisting of WSJ 90-92.

We first define empirically two fixed RSV thresholds Thi and Tlo. Documents with RSV>Thi should have high probability of being relevant to a query, and the opposite is true for documents with RSV<=Tlo. Based on our experience with PIRCS in general we set Thi = 2 or 3 and Tlo = 1.3 or 1.6 depending on whether the objective utility is lax or strict. At the start, the current RSV threshold T was set to Thi in order to improve the chance of getting relevant documents for learning. This contrasts with more realistic filtering situations where one might set a low initial threshold in order to return some documents for the user in order to avoid null answers.

Once the process starts, statistics of term usage is kept for all documents filtered, as well as separate term usage statistics for documents that were selected and found relevant. We did not store any document ID's, their individual or textual properties. In addition, a running total of the number of documents N that pass through the system, the number examined Ne, and the number found relevant Nr are also kept. This allows us to evaluate an overall average precision Nr/Ne for the user and the proportion of documents examined Ne/N at any time.

To reduce the complexity of the problem we decided to implement query weight adjustment only, and not query expansion. The starting topic descriptions are paragraphs in length augmented with concepts and are very rich already. We conjecture that not doing query expansion may not be too much of a loss. Moreover, the RSV's of documents would be in a more stable range. The update schedule is arbitrarily set to once every 10,000 documents filtered so as to be more efficient. Since there is a total of about 323K documents, we updated about 30 times. The weight of each of our query term is defined by log (p/(1-p))*((1-q)/q), where p = Pr(term present|Relevnat) and q = Pr(term present|~Relevant). These p and q values are modified using the captured term statistics during each update.

We try to dynamically adjust the RSV threshold T (to determine select or not select a document) based on N, Nr, and Ne. Specifically:

```
if (no relevants seen yet)
        T = T*x1 when T >Tlo & Ne/N<SRT
else if (Nr/Ne < Fi  & Nr > Nd) T = T*x2
else if (Nr/Ne < Fi & Nr <= Nd) T = T*x3
else T = T*x4 when Nr <= Nd
where x1=0.95; x2=1.03; x3=1.005; x4=0.97.
```

SRT (selection rate threshold) is set to 0.001 to prevent relaxing T too much if there is too much selection already but no relevant document has been obtained. Fi relates to the utility objectives and is set to 0.4 and 0.2 when i=1,2. Nd is a relevance document threshold (set to 30) used to control against too much restriction in T when not too many relevant documents have been seen yet. The parameters are biased towards a tighter condition than Fi.

Results & Discussion

Table 6a,b summarized the results of the adaptive filtering runs which are named 'pirc8FA1' and 'pirc8FA3' respectively for the two utilities F1 and F3. F1 aims at selecting all documents with a probability of relevance > 0.4, and for F3 it is a much more relaxed 0.2. The tables

	Comparison with Median			**F1 Utility**		**prec/rec**		**#of qrys**	
	>	=	<	score	docs	P	R	N+/0	N-
AP88	37,8	2	11	-93	669	.372	.097	34	16
AP89	21,6	4	25,3	-147	1841	.384	.214	20	30
AP90	16,7	4	30,3	-918	1304	.259	.170	18	32
AP89-90				-1065	3145	.332	.198	19	31

Table 6a: Adaptive Filtering F1 Results

	Comparison with Median			**F3 Utility**		**prec/rec**		**#of qrys**	
	>	=	<	score	docs	P	R	N+/0	N-
AP88	27,3	6	17	982	3748	.252	.369	27	23
AP89	24,5	3	23,3	2900	5245	.311	.493	26	24
AP90	22,9	5	23,5	756	3729	.241	.452	24	26
AP89-90				3656	8974	.281	.478	26	24

Table 6b: Adaptive Filtering F3 Results

show comparison with the median of all runs, the F score, number of documents selected, precision and recall values, and the number of queries with >=0 scores and negative scores. Table 6a shows that results for AP88 compares favorably with the median, drops some for AP89 and precipitously for AP90. On closer examination, it turns out that the parameter Tlo was not set right: it was left at 1.3 from previous runs, rather than the higher 1.6 value. It appears many documents with RSV's between 1.6 and 1.3

are allowed to be selected and these have relevance density much less than 0.4. When we repeated the runs with Tlo set to our original 1.6 value, the F1 scores become -93, 172, & -437 and the documents selected become 669, 1579 and 976 respectively for AP88, 89 and 90. The scores are substantially better, but still in the negative score territory for AP90. Table 6b shows the same runs for utility F3. Our runs achieve positive utility and compares favorably with median for all years.

5. Routing Retrieval and Batch Filtering

5.1 Genetic Algorithms

Genetic Algorithms (GA) [Holl75,Gold89,Davi91,Mitc96] are search procedures based on survival of the fittest. A general description of GA may be summarized as follows:

```
Procedure GA
begin
        initialize population P(0)
        evaluate P(0)
        t=1
        repeat
                select P(t) from P(t-1)
                recombine P(t)
                evaluate P(t)
        until (termination condition)
end
```

We utilize the TREC7 data as a test-bed for our research in applying genetic algorithms learning for Information Retrieval in conjunction with our PIRCS system.

To apply genetic algorithms to a specific application domain, one must devise a representation of the potential solution. Then we start the competition between the various potential solutions as specified by the genetic algorithm. The arbitrator of the competition is the fitness function, which is also application specific.

5.1.1 Representation of IR for GA

We describe four types of representations for GA-based approaches to information retrieval below.

1) A retrieval system assigns various weights for its retrieval function. Most frequently, weights are assigned to terms based on various statistics or learning algorithms. A GA can search the solution space by multiplying these weights by 0 or 1. This is equivalent to performing selection of a subset of the original terms.

2) If we allow the multiplier to be a real number in the range [0,1] in the previous scheme, then the weights will be modified and the GA performs a weight optimization.

3) Assuming a retrieval system generates a query with N given training documents, then a subset of these documents may generate a different query. Allowing the GA to search this space will often result in a superior query. This scheme can be further refined, by assigning weights to training documents. This is similar to bagging and boosting [Brei96,FrSc96,Quin96].

4) Given retrieval system r, which assigns a Retrieval Status Value (RSV_r) to retrieved documents, the output of different retrieval systems can be combined by some function f (RSV_r). A simple function of this type is the linear sum_of(a_r * RSV_r) where the a_r are arbitrary weights. A GA can search this space to yield a superior combination. This GA performs a ‘second level’ retrieval.

5.1.2 GA Fitness Functions for IR

The second domain specific component that has to be defined for the GA is the fitness function. A good fitness function should reflect the quality of retrieval. We describe three possibilities below.

1) Gordon [Gord88] and Chen [Chen95,ChSS98] have used the Jaccard similarity measure in previous GA research in IR. This measure is related to the number of retrieved relevant records and does not account for ranking.

2) The maximum likelihood measure is used for logistic regression. The advantage of using this fitness function is that it yields an RSV that corresponds to the probability of the document being relevant.

3) Average uninterpolated precision is the evaluation measure used by TREC. Since this is the measure one really tries to optimize, it is a logical choice for a fitness function.

5.2 Routing Retrieval Track

5.2.1 Routing query Pirc8R1

Pirc8R1 is a PIRCS retrieval based only on the query text and the term statistics of the AP88 collection without any training. It serves as a baseline measure for the effectiveness of other retrievals.

5.2.2 Routing query Pirc8R2

Pirc8R2 is a ‘second level’ retrieval by a GA, described in Section 5.1.1 as the 4th possible representation type. Nine retrievals were combined. They are described below:

a) Retrieval with no training documents (not1 and not2)

The PIRCS retrieval system itself is a combination of two networks: one is called query-focused retrieval and the other is called document-focused retrieval depending on the direction of activation spreading. Usually the user is allowed control over the combination coefficient to fine tune retrieval effectiveness. Here, we let the GA choose the coefficient. These two retrievals will be referred to as not1 and not2.

b) Standard PIRCS retrievals (pir1 and pir2)

We similarly create two retrievals using the standard PIRCS retrieval engine. All relevant documents from the AP88 collection were used for the training. Term expansion was set to 250 terms. These two retrievals are called pir1 and pir2.

c) GA used for term selection (gt1)

This is 1st representation for IR described above. The evaluated relevant and not relevant documents were divided into two groups. One group was used to select 300 candidate terms using the standard PIRCS term selection algorithm. The second group was used to train the GA. On the average about 1/3 of the terms were selected.

d) GA used for weight optimization (gw1 and gw2)

This is 2nd representation for IR described above. The evaluated relevant and not relevant documents were divided into two groups. One group was used to select 100 candidate terms using the standard PIRCS term selection algorithm. The second group was used to train the GA. The resulting queries with modified weights were used to yield separate retrievals.

e) Pure GA retrieval (ga)

The previous GA was used to optimize an existing retrieval system using the average uninterpolated precision as their evaluation measure. We also experimented with a pure GA, which uses the maximum likelihood measure. For terms we use about 300 terms, about 100 term pairs and also term triples and quadruples. Since all processing and evaluations are done internally and faster, we were able to use larger populations and more generations than in the previous cases. However we were not able to retrieve from the full AP89/90 collections, so the final retrieval was on 1500 documents selected by the standard PIRCS system.

f) Backpropagation network (bp)

The final retrieval is a backpropagation neural network. We modified NevProp, a publicly available c program maintained by Phil Goodman of the University of Nevada. The terms, test data and retrieval were done in the same way as the pure GA retrieval.

Details of the nine retrieval methods are summarized in Table 10 at the end of this paper.

Selecting training documents for the second level retrieval required some compromises. The ideal situation would be if unseen documents were used, since second level retrieval attempts to predict the behavior of these retrievals for new documents. However many of the component training methods were exposed all training documents. Some of the choices we considered were:

1) Reserve some portion of training data for second level retrieval only. This was deemed unpractical due to the small number of training data for many queries.
2) The only method trained on all training data was the pir1 and pir2. The other methods used half of the training data for term selection only. Create a version of pir1 and pir2 using only half the data to find the combination function and use the fully trained version for the actual addition. In this case too some queries would have inadequate number of training data.
3) Same as 2) but use all training documents. This was the method selected, so that we can use a uniform method for all queries.

5.3 Routing Retrieval Results

Table 7 shows our results in comparison with the median. The combined retrieval Pirc8R2 obviously performed much better than the retrieval without training Pirc8R1. Considering that not all available judged documents were used, only those from AP88, the results are quite good.
Table 8 shows performance of the various methods for the 43 queries, which had both relevant training and testing documents. The method column contains the 9 methods, the 2 official routing runs and max which is the retrieval we get if we choose the best performing method for each query. As can be seen the individual retrievals did not do well. They all performed worse than not2, the better of the retrievals with no training, except for pir1 and pir2, the classic PIRCS retrievals. However the second level retrieval (pir98r2) was successful. It improved 44% over not2, 17% over pir1 and was only 1% worse than max, which is a hypothetical combination that can only be done if results are available.

A possible explanation is the lack of adequate number of training documents. If we just look at the 13 queries, which had the most relevant training documents, the results are

	Pirc8R1			Pirc8R2		
	>	=	<	>	=	<
avg prec	11(3)	2	36	30(7)	7	12

Table 7 Comparison of routing results with median. Number in parenthesis is number of best values.

different. All training methods are better than no training, although still not as good as pir1. Also the second level retrieval outperforms max.
Apparently the 'second level' retrieval is not that much affected by the lack of training documents as the other methods, since its dimensionality (9, the number of retrievals combined) is much lower than the dimensionality of the other methods (>100, the number of terms).

The implication of this experiment is, that it may be quite possible to achieve a retrieval that is close to the maximum of all TREC7 retrievals by a genetic algorithm based second level retrieval.

method	*43 qry*	*% imprv over not2*	*13 qry*	*% imprv over not2*
not1	0.2699	-5.7%	0.2626	-17.7%
not2	0.2862	0.0%	0.3193	0.0%
pir1	0.3519	23.0%	0.4531	41.9%
pir2	0.3341	16.8%	0.4351	36.3%
gw1	0.2705	-5.5%	0.4402	37.9%
gw2	0.2724	-4.8%	0.4118	29.0%
gt1	0.2320	-18.9%	0.3533	10.7%
np	0.2619	-8.5%	0.3676	15.1%
ga	0.2536	-11.4%	0.3614	13.2%
max	0.4172	45.8%	0.5021	57.3%
pir98r1	0.3145	9.9%	0.3721	16.5%
pir98r2	0.4136	44.5%	0.5208	63.1%

Table 8: Individual retrieval results.

5.3 Batch Filtering Track

The Pirc8R2 query was used for batch filtering for the 44 queries, which had relevant training documents. For the others 6 queries, no documents were submitted for the F1 measure. For F3 measure, if the RSV was larger then the largest RSV for Pirc8R1 retrieval then they were selected. Previously we used the maximal F measure point to select the cutoff. This time we utilized a logistic regression transformation. (using a GA). In the past we found that performance in the batch-filtering task was highly correlated with performance in routing, therefore the average performance shown above is a good sign. Table 9 below shows how we compared with the median.

run	>	=	<
Pirc8FB1	23 (11)	17 (10)	10
Pirc8FB3	26, (8)	16 (8)	8

Table 9: Comparison of batch filtering results with median. Number in parenthesis is number of best values.

6. Conclusion

Our system continues to perform well for the ad-hoc short queries. Adaptive filtering was done for the first time. Our simple approach of adjusting thresholds that does not involve storing large amounts of past data seems to work reasonably well. We also show that combination of retrieval based on a genetic algorithms approach can provide superior results in both routing and batch filtering.

Acknowledgments

This work is partially supported by an U.S. Department of Defense contract MDA904-96-C-1481.

References

[Brei96] Breiman, L. (1996) Bagging predictors. Machine Learning, 24: 123-140.

[Chen95] Chen, Hsinchun (1995) Machine Learning for information retrieval: Neural networks, Symbolic Learning and Genetic Algorithms. JASIS 46(3) 194-216.

[ChSS98] Chen, Hsinchun, Shankaranarayanan, G, She, L (1998) Machine Learning approach to inductive query by examples: An experiment Using Relevance Feedback, ID3, Genetic Algorithms, and Simulated Annealing. JASIS, 49(8) 693-705.

[Davi91] Davis, L. in Handbook of Genetic Algorithms; Davis, L. (ed.); Van Nostrand Reinhold: New York, 1991

[FrSc96] Freund Y, Schapire R,E. (1996) A decision theoretic generalization of on-line learning and an application to boosting ("http:/www.research.att.com/orgs/ssr/people/schapire")

[Gord88] Gordon, M. (1988) Probabilistic and genetic algorithms for document retrieval. CACM 31(10), 1208-1218.

[Gold89] Goldberg, D.E. Genetic Algorithms in Search Optimization & Machine Learning. Addison-Wesley, 1989.

[Holl75] Holland, J.H. Adaptation in Natural and Artificial Systems. University of Michigan Press: Ann Arbor, MI, 1975.

[Kwok96] Kwok, K.L. (1996) A new method of weighting query terms for ad-hoc retrieval. In: Proc. 19th Ann. Intl. ACM SIGIR Conf. On R&D in IR. Frei, H.P., Harman, D. Schauble, P. & Wilkinson, R. Aug 18-22, 1996. Pp.187-195.

[KwCh98] Kwok, K.L. & M. Chan (1998) Improving two-stage ad-hoc retrieval for short queries. In: Proc. 21st Ann. Intl. ACM SIGIR Conf. On R&D in IR. Croft, W.B, Moffat, A, van Rijsbergen, C.J, Wilkinson, R & Zobel, J. Aug 24-28, 1998. Pp.250-256.

[Mitc96] Mitchell, M. An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, 1996.

[Quin96] Quinlan, J.R (1996) Boosting, Bagging and C4.5, In Proceedings, Fourteenth National Conference on Artificial Intelligence, 1996.

[SiMB97] Singhal, A, Mitra, M & Buckley C (1997). Learning routing queries in a query zone. In: Proc. of 20th Ann. Intl. ACM-SIGIR Conf. On R&D in IR. Belkin, N.J, Narasimhalu, D, & Willett, P (eds). pp.25-32.

[StCM95] Strzalkowski, T, Carballo, J.P & Marinescu, M (1995). Natural language information retrieval: TREC-3 report. In: Overview of the Third Text REtrieval Conference (TREC-3). Harman, D.K. (Ed.). NIST Special Publication 500-225, pp.39-53.

[WRBJJ98] Walker, S., Robertson, S.E. & Boughanem, M., Jones, G.J.F & Sparck Jones, K (1998). Okapi at TREC-6 automatic ad hoc, VLC, routing, filtering and QSDR. The Sixth Text REtrieval Conference (TREC-6). Voorhees E.M & Harman, D.K. (Eds.). NIST Special Publication 500-240, pp.125-136.

[XuCr96] Xu, J. & Croft, W.B. (1996). Query expansion using local and global document analysis. In: Proc. 19th Ann. Intl. ACM SIGIR Conf. On R&D in IR. Aug 18-22, 1996. Frei, H.P., Harman, D. Schauble, P. & Wilkinson, R. pp.4-11.

Method	Training data	Term selection	Avg. number of terms	Training method	Retrieval
(not1)	none	All terms from query subject to Zipf cutoff [3,10000]	32	Self training based on statistics	Full ap89 and ap90
(not2)	none	same	same	same	same
(pir1)	All relevant	Term expansion ~250	220	Pircs network	same (only for queries with training docs)
(pir2)	same	Term expansion ~60	60	same	same
(gt1)	Half for term selection Other half for training	Ga selects From 300 terms	90	Ga selects terms for pircs	same
(gw1)	same	Term expansion ~100	80	Ga adjusts weights for pircs	same
(gw2)	same	Term expansion ~100	80	same	same
(ga)	same	~300 terms 100 doubles triples and quadruples	410	Genetic algorithm	1500 documents retrieved by pir1
(bp)	same	same	same	backprop	same

Table 10: Summary of Nine Retrieval Methods for Routing and Batch Filtering