TREC-8 Ad-Hoc, Query and Filtering Track Experiments using PIRCS

K.L. Kwok, L. Grunfeld and M. Chan
Computer Science Department, Queens College, CUNY
Flushing, NY 11367

Abstract

In TREC-8, we participated in automatic ad-hoc retrieval as well as the query and filtering tracks. The theme of our participation is 'retrieval lists combination', and the technique is applied throughout our experiments to various degree. It is pointed out that our PIRCS system may be considered as a combination of probabilistic retrieval model and a language model approach. For ad-hoc, three types of experiments were done with short, medium and long queries as before. General approach is similar to TREC-7, but combination of retrieval lists from different query types were used to boost effectiveness. For query track, we submitted one short-query set, and performed retrieval for twenty one natural language query vairants. For filtering track, experiments for adaptive, batch filtering, and routing were performed. For adaptive, historical selected document list was used to train profile term weights and dynamically vary retrieval status value (rsv) threshold for deciding document selection during the course of filtering. For batch filtering, Financial Times FT92 data was used to define 6 retrieval profiles whose results were combined based on coefficients trained via a genetic algorithm. Logistic regression transforms rsv`s to probabilities. Routing was similarly done with additional training data obtained from non-FT collections and two additional profiles were defined and combined

1. Introduction

We continue to use our PIRCS system for investigation. A theme that we emphasize this year is 'retrieval combination'. Given an information need different query formulation or different search algorithms may retrieve quite different document sets. Combining their retrieval status values (RSV) may reinforce common relevant ones and lead to new ranking that is more effective than the original separate sets. The idea has been in existence in IR practice and literature, and proposed by many people for many years. We employ it to various degree as a way to refine our various experiments.

The basic PIRCS system is a combination of two retrieval algorithms: document-focused and query-focused. In Section 2, we point out that document-focused weighting is similar to weighting based on a language model approach.

In addition to combination, two strategies for ad-hoc are: 2-stage retrieval and collection enrichment as done in TREC-7. Both strategies have been found to work more often than not for queries of different lengths. Ad-hoc retrievals are discussed in Section 3.

In query track, we use our system with 21 variants of topics numbered #51-100 to retrieve on Disk 1, and some observations of the results are given in Section 4.

In the filtering track, adaptive filtering was done by using accumulated selected documents to help set RSV thresholds for future document selection. Batch filtering makes use of FT92 known data to help train multiple variant profiles and their (near) optimal combination coefficients. These were used to simulate final filtering on FT93 & 94 without adaptation. In routing additional profiles, coefficients and training data were used to produce ranked outputs. Adaptive filtering is described in Section 5, batch filtering in Section 6, and routing retrieval in Section 7. Section 8 contains our conclusions.

2. PIRCS Weighting and Language Model

Given a query q to retrieve documents d from a collection, our basic PIRCS system itself is a combination of two retrieval algorithms producing a document-focused and a query-focused RSV's for each document with a mixing parameter α. Thus (see [Kwok95] for greater details):

$$RSV(q,d) = \alpha * RSVd + (1-\alpha) * RSVq \quad (1)$$

with

$$RSVd = \Sigma_k S(qtf_k/L_q) * w_{dk} \quad (2a)$$

$$w_{dk} = \log [tf_k/(L_d - tf_k) * (Nw - L_d - F_k + tf_k)/(F_k - tf_k)] \quad (2b)$$

and

$$RSVq = \Sigma_k S(tf_k/L_d)^* w_{qk} \quad (3a)$$

$$w_{qk} = \log [qtf_k/(L_q - qtf_k)^*(Nw-F_k)/F_k] \quad (3b)$$

where tf_k, qtf_k are the frequency of term k in d and q respectively, $L_d = \Sigma_k tf_k$, $L_d = \Sigma_k qtf_k$ are the lengths of d and q, S is a sigmoid-like function, $F_k = \Sigma_{all\ doc} tf_k$ is the collection frequency of term k, and $Nw = \Sigma_k F_k$ is the number of tokens used in the collection.

Our approach considers a document (or query) as constituted of conceptual components approximated as single terms and self-relevant to the document (query) itself, and we work in a universe consisting of document components rather than documents. Because of the self-relevance assumption, every query (document) therefore has a relevant and irrelevant set even when no relevant judgment has been made, we are able to bootstrap and provide probabilistic weights to our terms at the initial retrieval stage. Because we work with conceptual components, repeat term usage and item lengths are accounted for, enabling us to remove the binary assumption restriction. The weighting of Eqn. 3b is the familiar probabilistic query term weights but in the component environment. Eqn. 2b is for document-focused retrieval and the form of the weighting, after taking the approximation Nw >> all other frequencies, turns out to be very similar to those used by [HiKr98] via a language model approach, but with a different smoothing coefficient.

Thus, our PIRCS system may also be viewed as a combination of the probabilistic retrieval model and a simple language model. For many of the past TREC experiments, our system has been demonstrated to provide superior effectiveness, and last year it was observed that PIRCS is one of few automatic systems that provides many unique relevant documents in the judgment pool [VoHa98]. We believe this is because our system is unique among participants in that it is a combination of two different models.

3 Ad-Hoc Retrieval

The target collection for ad hoc retrieval is from Disks 4&5, consisting of articles from Financial Times, Federal Register, Foreign Broadcast Information Service and the LA Times, some 2 GB of text in over ½ million documents. These are similar to TREC-7 and we used last year's processed data unchanged.

TREC-8 topics are described in several sections: title, description and narrative. This year, the official ad-hoc runs should make use of the title and description sections only. We call this run pir9Attd. It is a combination of retrieval lists from pir9At0 (title only) and pir9Atd0 (title and description). We consider this group to be short to medium queries. In addition we have two more submitted runs called pir9Aa1 and pir9Aatd, the former has queries making use of all sections, and the latter combines an all section run with pir9Atd0. The title, title+description, and all section queries have on average 2.54, 6.14 and 12.8 unique terms respectively after stemming and stopword removal.

Results for short and medium queries are discussed in Section 3.1 and long queries in Section 3.2.

3.1 Short and Medium Queries

We follow our TREC-7 approach to short query retrieval by using five methods successively to produce a final retrieval list. These five methods [KwCh98] are: 1) average within-document term frequency to weight short query terms (avtf query term weighting); 2) variable high frequency Zipfian threshold dependent on query size; 3) collection enrichment to improve initial stage output relevant density; 4) enhancing term variety in raw queries by adding highly associated terms from initial retrieval based on mutual information measure; and 5) using retrieved document local term statistics to improve weighting conditioned on irrelevancy in final retrieval.

	Query Type submit`d		submit`d		official	
	pir9At0 value	% inc	pir9Atd0 value	% inc	pir9Attd value	% inc
Relv.Ret	3299	0	3272	-1	3342	1
Avg Prec	.3063	0	.3022	-1	.3207	5
P@10	.4800	0	.4920	3	.5080	6
P@20	.4410	0	.4290	-3	.4450	1
P@30	.4027	0	.3807	-5	.4033	0
R.Prec	.3326	0	.3301	-1	.3441	3

Table 3.1a: Automatic Ad Hoc Results for 50 Short and Medium Queries

	Query Type unsubmit`d		official		submit`d		<corrected runs >			
	pir9Aa0 value	% inc	pir9Aatd value	% inc	pir9Aa1 value	% inc	pir9Aa1* value	% inc	pir9Aatd* value	% inc
Relv. Ret	3344	0	3382	1	2751	-18	3350	0	3383	1
Avg Prec	.3241	0	.3303	2	.2624	-19	.3249	0	.3322	3
P@10	.4940	0	.5120	4	.4500	-9	.5040	2	.5140	4
P@20	.4530	0	.4600	2	.3860	-15	.4470	-1	.4610	2
P@30	.4080	0	.4120	1	.3327	-18	.4033	-1	.4107	1
R.Prec	.3441	0	.3512	2	.3065	-11	.3421	-1	.3515	2

Table 3.1b: Automatic Ad-Hoc Results for 50 Long Queries

For collection enrichment, we form a miscellaneous collection by retrieving the top 200 documents from the sub-collections AP1-3, WSJ1-2, FB6 and using the title form of the queries. This miscellaneous collection is used to enrich the top-ranked set of the initial stage retrieval from the target collection. This year we modified the method slightly by limiting the number of external documents for feedback to a maximum so as not to overwhelm expansion based on documents from the target. It helps for TREC-7 but is slightly worse for TREC-8. We also employ combination of retrieval lists to help improve effectiveness; coefficients of combination are learnt from TREC5 to 7 results.

Results and Discussion

Our TREC-8 results for short and medium queries are summarized in Table 3.1, and their nomenclature has been described in the Introduction. The title only (t0: mean av. prec. 0.3063) and title+description (td0: mean av. prec. 0.3022) runs are very close, with a slight edge to the former. This year there are several highly specific topics with words like 'osteoporosis' #404, 'Schengen agreement' #410, 'killer bee attacks' #430, 'supercritical fluids' #444. They are better with the title alone than adding the description. Title only has 26 queries with better average precision, 19 worse and 5 equal to title+description. However, for retrieved relevants at 1000, the numbers are reversed: 14:15:21. Longer queries generally tend to get better recall as was also found in our previous TRECs. Best result is obtained by combining their retrieval lists (ttd) giving improvements of about 5% over (t0), and is our designated official run. It also has a relevant retrieved at 1000 documents of 3342 which is about 70.7% of the pooled documents that have been judged relevant (4728).

Comparisons with the all-sites median average-precision, precision at 100 and 1000 documents are given in Table 3.2. Our runs are well above median. For example, the official combination run (ttd) has average precision better than median in 43 instances with 3 queries achieving the best, and are worse than median in 7 cases. For title only (t0), the number of queries with precision better, equal or worse than median are: 35:4:11. Out of the 35, 11 have the best values. This year the title only and title+description medians are evaluated separately.

			official		official
	pir9At0 > = <	pir9Atd0 > = <	pir9Attd > = <	pir9Aa1 > = <	pir9Aatd > = <
AvgPrec	35,11 4 11	34,1 1 15	43,3 0 7	24,1 0 26	37,6 3 10
RR@100	34,18 8 8	36,5 4 10	41,9 3 6	22,7 6 22	36,11 4 10
RR@1K	33,21 11 6	38,14 7 5	40,14 5 5	18,9 12 20	39,18 7 4

Table 3.2: Ad-Hoc Results: Comparing All Submitted Runs with Median

Year to Year Comparison

This year's very short (title only) ad-hoc effectiveness is much better than TREC-7, and deserves some discussion since both years use the same collection. Within our site for example, last year's MAP (mean average precision) was 0.2427 for the title only run and 0.3063 this year. The MAP difference of 0.0636 (over 20% improvement) does not seem explainable by parameter adjustments alone. The reason is because the topics for this year appears much easier. If we use a value of AP => 0.5 as an indicator of easy topics, then there are 8 this year and only 4 in TREC-7. These 8 and their key terms are: 403 (osteoporosis), 410 (Schengen), 415 (golden triangle), 420 (carbon monoxide), 423 (milosovic), 430 (killer bee), 441 (Lyme), and 444 (supercirtical fluid). The AP sum of this 8 totals 5.6507. The sum of last year's 4 easy topics plus the next 4 top totaled 4.5988. This estimated difference amortized over 50 queries contributes .0210 to the MAP difference, or 33% of the observed increase. Thus, in year to year comparison, as we already noted in TREC-7, topic hardness can play a substantial rule.

3.2 Long Queries

Long queries can use all sections of a topic. Our official long query run is pir9Aatd, which is a combination of the title+description only run pir9Atd0 and another that uses all wordings of a topic pir9Aa0 (un-submitted). In addition, we submitted another run called pir9Aa1, which is the basic pir9Aa0 with phrase-reranking added. Unfortunately, an error was committed during the phrasing operation. Each topic content was first POS-tagged and each sentence was broken down into noun phrases. A choice can be made to keep only the noun phrases or to keep the residual entries such as verbs, adverbs as well. The wrong choice of keeping more than noun phrases was made. This leads to erroneous re-ranking of the retrieved documents and bad results for pir9Aa1. After results were known, we re-do the pir9Aa1 run correctly (now called pir9Aa1*), and these are tabulated in Table 1b. By some fortune, our pir9Aatd combination run was done using the uncorrupted pir9Aa0, and it gives very good results. Had we combined pirc9Atd0 with the corrected pir9Aa1* run, the result would be slightly better as shown under the pir9Aatd* column in Table 3.1b.

It is seen from Table 3.1b that phrase re-ranking in pir9Aa1* (with mean average precision of 0.3249) does not do much to pir9Aa0 (0.3241). Combining the all section run with the title+description run can lead to about 2-3% improvements over the components. When compared to results from all sites, Table 3.2, our official

long query run has 40 queries equal or above the median with 6 of them being best, and 10 queries worse than median.

4. Query Track

The purpose of the query track is to explore how query variants of the same topical concepts may affect retrieval results. Topics used are 51-100 and retrieval was done on Disk 1. The data consists of variations in average non-interpolated precision for three dimensions T, Q and R. T represents different topics. Q denotes different query compositions for each topic (total of 23 query variants, 21 of which are natural language type and one is our pir1a. Two more are weighted query types which we did not analyze. 'pir' denotes our system. The '1' in '1a' means very short version; longer versions like a sentence are denoted by '2' or '3'). R means different retrieval algorithms (specifically 8 of them like INQ, Sab, etc; one of which pir is our PIRCS system). Readers are advised to refer to the query track report for a description of these queries and retrieval algorithms.

For comparison we will use the average non-interpolated precision. We first try to see which query type does best for each topic using our PIRCS retrieval engine by noting the best retrieval within each topic (i.e. R=pir, for each T, find best Q). It shows query type Sab1c performs best 9 out of 50 times, and a group of 6 other query types (INQ1c, INQ2e, INQ3d, Sab1b, Sab3a and pir1a) perform best 4 out of 50. Others have less. It seems the Sab1c query formulation agrees well with our engine. When we evaluate the average precision over all topics for each of the 21 query type using our engine (i.e. R=pir, for each Q average over all T), our pir1a formulation returns the best performance at 0.3030. Putting this in perspective, the title section of the TREC original topics gives an average precision of 0.2973. If the title, description and narrative sections are used to produce long queries, the average precision is 0.3330.

When the data is averaged over 21 query types for all 50 topics (i.e. for each R, average over all Q and T), we can see how each retrieval algorithm performs. It seems that our pir method returns an average precision of 0.2458, practically the same as Sabe's 0.2459.

When the data is averaged over 21 query types and all 8 algorithms (i.e. for each T, average over all Q and R), one may get some idea of how hard each topic is for retrieval. Average precision varies from 0.6527 (topic 70) to 0.0131 (topic 74). Unlike the current ad-hoc experiments, there are no topics with highly specific terms like `oesteoporosis` or `Schengen agreement` that can return precision values of 0.8 or higher. Topics 58 (0.5640) and 59 (precision 0.0988) seem to represent one easy and one hard topic, and we choose them to have a closer look.

Since we cannot run other retrieval algorithms, we focus on the results for topic 58 and 59 returned by our pir engine. For topic 58, out of 21 query types, only 5 have average precision of less than 0.6884, showing that it is an easy topic. The reason these 5 do not do well is because the words `rail strike(s)` were not used in their formulation. Instead, `railroad strike`, `strikes .. against railroads`, or `labor relations .. in transportation industry` are used.

	query	Av.Prec. Initial	Av.Prec. Final
1	rail strikes (has 'railstrike') - NIST title	0.6872	0.7537
2	rail strike reports ('') - INQ1a	0.6858	0.7497
3	rail strikes, walkouts ('') - pir1a	0.6413	0.7364
4	strikes by rail (no 'railstrike')	0.6755	0.7347
5	NIST long query (has 'railstrike')	0.5393	0.7173
6	railstrikes	0.2133	0.6915
7	rail walkouts	0.3957	0.6754
8	railway strikes	0.2704	0.3908
9	railroad strikes	0.3165	0.2946
10	Line1 (.5) combine Line7 (.5)		0.7414
11	(.7) (.3)		0.7534
12	(.8) (.2)		0.7556

Table 4.1a: Topic #58 - Average Precision for Different Query Variants

Table 4.1a shows some deliberate variations of wordings for Topic 58. For human understanding, `rail strike`, `railroad strike`, and `railway strike` seem synonymous; yet for retrieval the latter ones are much worse (Lines 1, 8 & 9, average precision of .7537, .3908, .2946 respectively). The juxtaposition of `rail strike` also contribute an additional 2-word phrase `railstrike` in our system, but its effect is small (Lines 1 & 4) and does not account for those large differences. Lines 2 & 3 shows the idiosyncrasy of IR: one would expect `walkouts` to add more content and focus to `rail strikes`, yet it is worse than adding `reports`. Paragraph size query is not as good as the two words in this case as shown in Line 5. Lines 7, 8 & 9 show that the term `rail` is critical for this query concerning strikes. But, how does one know during query formulation? The document frequency of rail, railroad and railway are: 3108, 3902, 1516 and do not seem able to indicate the usefulness of one or the other. Perhaps one may say that `rail strike(s)` is the normal description of this concept.

In the spirit of our theme, we try combining queries of Line 1 & 7 giving results in Lines 10, 11 & 12. With the right coefficient, it can surpass the best of its constituents. Even choosing a coefficient of 0.5 is not

bad at 0.7414 average precision, better than using the UNION of such words in a single query shown in Line 3.

Topic 59 is one kind of hard topic, with some of its variant results shown in Table 4.1b. Out of 21 query types, only 8 achieve precision above 0.1 (best is Line 1 of 0.3420) according to our pir engine. It asks for a very

	query	Av.Prec Initial	Av.Prec Final
1	storm deaths - Sab1b	0.2255	0.3420
2	what damaging weather events have caused deaths - INQ2b	0.1205	0.3206
3	storm fatalities	0.0869	0.3147
4	has violent rain storms caused many deaths - INQ3d	0.3173	0.3099
5	deaths caused by storms s/as typhoons, hurricanes, tornados- Sab3a	0.3098	0.2744
6	weather deaths, injuries - pir1a	0.0699	0.2718
7	NIST long topic	0.0157	0.0060
8	weather related fatalities - NIST title	0.0218	0.0267
9	Line1 (.5) combine Line6 (.5)		0.4253
10	(.7) (.3)		0.4197
11	(.8) (.2)		0.3930
12	Line1 UNION Line6	0.2709	0.4035

Table 4.1b: Topic #59 - Average Precision for Different Query Variants

general concept ‘weather related fatalities‘. Two simple words ‘storm deaths‘ is best for this retrieval (Line 1), while ‘storm fatalities‘ (Line 3) is also very good. It turns out that ‘fatalities‘ without ‘storm‘ is a bad choice. Queries using ‘weather‘ with ‘fatalities‘ all return miserable results like (Line 8). ‘deaths‘ seem to be a more effective choice (Line 2 & 6) although it is difficult to see why one is better than the other at query formulation. ‘Weather related‘ is very general, and it would seem spelling out the more common occurring specific cases such as: hurricanes, tornadoes, floods, rain etc. may be more useful. Line 4 did just that but surprisingly it was only good (0.2744) and not the best. The word ‘storm‘ seems to capture the concept ‘weather related bad things‘ well as it is less polysymous than ‘weather‘. Combination of retrieval lists (Lines 9, 10 & 11) or combining terms in one query i.e. longer query (Line 12) can boost effectiveness substantially even for this hard query.

The study of these two queries only shows that the choice of words for retrieval is crucial for good results. How to make a good choice is not at all clear.

5 Adaptive Filtering Track

This year‘s adaptive filtering task makes use of topics #350-400 to select documents from the FT (Financial Times) collection from 1992 to 1994 in date order. Adaptive filtering is difficult. A possible approach is to use a two step strategy: at start when little knowledge is known, a simple adaptive threshold-adjustment and profile re-weighting method is used. After sufficient relevant data is available, train and expand profiles carefully and do filtering without adaptation like in batch filtering. Batch filtering is discussed in Section 5.

To prepare for filtering, a dictionary was defined by processing some 1.2 GB of texts consisting of FT91, AP3, all of Foreign Broadcasting FBIS and WSJ-2 collections. These were chosen to be close to the time period 1992-94 as well as content. The dictionary size after stopword removal and Zipf thresholding is about 240K. The filtering collection FT92-94, with long documents segmented into sub-documents, were then processed against this dictionary with no manual classification codes used, only the text portion of each document. The setup was employed to debug codes for mapping physical document order on CDROM to given date order, but not used for training. Training was done using the TREC-7 AP filtering collections, and parameters transferred to this TREC-8 task. We corrected some bugs in our TREC-7 program and also modified our approach.

Many considerations are needed for adaptive filtering. These include defining an initial profile together with an initial selection threshold to start the process, adaptively train the profile to tailor to the type of documents seen so far, dynamically adapt the threshold to select or not select a document for examination, determine how often these changes are to be made, and at the same time attempt to maximize a target utility value. Both adaptation of the filtering profile and that of the threshold are useful. Improved profiles help to separate relevant documents from irrelevant ones better, based on probability or RSV values assigned. Threshold adjustments help to achieve a utility target for the selected documents. Our approach emphasizes on threshold adaptation. Threshold is adjusted periodically after a number of documents have gone through the process and when profiles are updated. Profiles are changed only when a new relevant document has been selected. Moreover, no query expansion was done.

Initial profiles are defined using the raw topic descriptions and our dictionary and term statistics. For document selection when no relevant documents are known, two RSV thresholds Thi and Tlo are defined initially. Documents with RSV>Thi should have high probability of being relevant to a profile, and the opposite is true for documents with RSV<=Tlo. These were set by calculating a profile self-retrieval RSV

(SRSV) [KwGL95]. Each profile is regarded as relevant to its own description when it is considered as a document, and this SRSV value is large. In reality, documents may only overlap partially with a profile and still be relevant, and their RSV's are much less than SRSV. Our two thresholds are defined as: Thi=hi*SRSV, Tlo=lo *Thi, where lo is fixed as 0.8, and hi depends on the utility target F. Typical hi values we used are 0.35 for F1 and 0.3 for F2 utilities. These values are based on experimentation with TREC-7 filtering discussed later. As filtering proceeds, Thi may be updated, but it is not allowed to fall below Tlo if no relevant documents have been selected.

Once the process starts, statistics of term usage is kept for all documents filtered. For documents selected, whether relevant or not, they are stored as a retrieval collection for training purposes. In addition, a running total of the number of documents N that passes 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 preg=Nr/Ne for the user and the proportion of documents examined Ne/N at any time. preg is a global precision indicator. In addition, a local Nr/Ne precision prel for the last two update rounds is also calculated for fine-tuning the adaptation of the threshold.

The update schedule is set to once every no=2,000 documents filtered based on experiments with the AP collection. We try to dynamically adjust the RSV threshold T (to determine select or not select a document) based on N, Nr, Ne. Specifically:

```
if (no relevants seen yet)
        T =T*(1-e) when T >Tlo & Ne/N<SRT
else {if (change in Nr) {
        update profile weights
        recalculate T using selected docs }
     if (change in Ne) {
        if (both preg & prel <G) T=T*(1+2*e)
        if (both preg & prel >G) T=T*(1-e) }
   }
```

SRT (selection rate threshold) is set to 0.001 to prevent relaxing T too much if there are too many documents selected already and none is relevant, e=0.05 is an adjustment rate. With other parameters fixed, we consider the utility performance as a function of G and hi. These are set to achieve maximal utility values according to training parameters from the AP collection as shown in Tables 5.1a,b. We submitted two runs for F1: pir9LF1 (hi =.35, G = .55) and pir9LF1a (hi = .35, G = .6), and two runs for F2: pir9LF2 (hi = .3, G = .4) and pir9LF2a (hi = .3, G = .45).

hi\G	.4	.45	.5	.55	.6
.3	-271	838	1342	1603	1661
.35	537	1040	1460	**1844**	**1672**
.4	431	835	1272	1268	1236

Table 5.1a: Training from AP collection - utility values as a function of hi & G: target F1=0.4

hi\G	.3	.35	.4	.45	.5
.27	2070	3028	3575	3590	3304
.3	2967	3823	**3772**	**3893**	3494
.35	2848	3314	3564	3346	2821

Table 5.1b: Training from AP collection - utility values as a function of hi & G: target F2=0.25

Results & Discussion

Table 5.2a,b summarize results of the adaptive filtering runs which are named pir9LF1 and pir9F1a respectively for the utility F1. F1 aims at selecting all documents with a probability of relevance > 0.4. In addition to F1 scores, we tabulate also docs (number of documents selected), #rel (number of relevant documents selected), precision and recall, and N+,o,- (number of queries that have positive, zero and negative utility). The two runs differ very little.

	Comparison with Median			**LF1**							
FT	>	=	<	score	docs	#rel	Prec	Recl	N+	No	N-
92	31,13	3	16,1	-575	520	.93	.179	.161	9	7	34
				(+278)		(576) best submitted					
93	14,9	15	21	-438	334	46	.138	.073	8	14	28
				(+260)		(629) best submitted					
94	27,15	6	17	-254	257	52	.202	.080	15	14	21
				(+260)		(647) best submitted					
92-4	27,8	2	21	-1268	1111	191	.172	.103	12	6	32
				(+494)		(1852) best submitted					

Table 5.2a: LF1 Adaptive Filtering for pir9LF1

	Comparison with Median			**LF1**							
FT	>	=	<	score	docs	#rel	Prec	Recl	N+	No	N-
92	32,13	2	16,1	-565	505	.89	.176	.155	10	7	33
				(+278)		(576) best submitted					
93	16,9	15	19	-429	332	47	.142	.075	10	14	26
				(+260)		(629) best submitted					
94	25,14	6	19	-261	253	49	.194	.076	13	13	24
				(+260)		(647) best submitted					
92-4	27,8	4	19	-1255	1090	185	.170	.100	13	5	32
				(+494)		(1852) best submitted					

Table 5.2b: LF1 Adaptive Filtering for pir9LF1a
Comparison

with Median LF2

FT	>	=	<	score	docs	#rel	Prec	Recl	N+	No	N-
92	12,5	11	27,8	-600	1092	123	.113	.214	13	4	33
				(+435)		(576) best submitted					
93	11,5	15	24,7	-174	438	66	.151	.105	16	9	25
				(+349)		(629) best submitted					
94	16,8	17	17,5	-39	251	53	.211	.082	16	10	24
				(+383)		(647) best submitted					
92-4	13,3	6	31,10	-813	1781	242	.136	.131	20	2	28
				(+990)		(1852) best submitted					

Table 5.2c: LF2 Adaptive Filtering for pir9LF2

Comparison with Median LF2

FT	>	=	<	score	docs	#rel	Prec	Recl	N+	No	N-
92	12,6	13	25,8	-583	1155	143	.124	.248	12	4	34
				(+435)		(576) best submitted					
93	9,5	13	28,9	-181	473	73	.154	.116	14	11	25
				(+349)		(629) best submitted					
94	15,8	15	20,6	-54	294	60	.204	.093	13	12	25
				(+383)		(647) best submitted					
92-4	10,3	6	34,12	-818	1922	276	.144	.149	17	3	30
				(+990)		(1852) best submitted					

Table 5.2d: LF2 Adaptive Filtering for pir9LF2a

This task was not successful as the F1 scores are negative for all years. The learning process for the profile weighting and threshold setting however seem correctly done as the scores get better in successive years. When compared with results from all participants, we have at least 29 instances better or equal to the median out of 50 for all years.

Tables 5.2c,d summarize our filtering runs for the LF2 utility target of 0.25 precision. As previously, utility scores improve year to year, but they are all negative, and results are below median. Filtering the FT collections appears quite a difficult task. Its characteristics seem very different from the AP collection; bringing parameters based on that collection seems not useful. Even the more restrictive parameters set for LF1 do not return positive scores for the LF2 target. However, after results were known, more restrictive parameters were set and we were able to achieve positive utilities of around 65 for F1 and 170 for F2.

6 Batch Filtering and Routing Retrieval

6.1 Pircs and genetic algorithms

The TREC8 filtering and routing tasks were used as a testbed for our research in applying genetic algorithms learning [Gold89,Holl75] in Information Retrieval, in conjunction with the Probabilistic Information Retrieval Component System (PIRCS). PIRCS itself is a combination of two networks, implementing different retrieval modes, query-focused retrieval (type 1) and document-focused (type 2) retrieval. The user is allowed control over the combination coefficients to fine tune retrieval effectiveness. If these coefficients are set to (0,1) and (1,0), the resulting retrievals will be virtually independent. There are other ways of getting differing retrievals, one of the most effective is varying the term expansion levels.

Given a 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 GA can search this space to yield a combination, which is superior to any individual retrieval. A simple function of this type, which we use in the current experiments is linear addition, $sum\_of(a_r * RSV_r)$, where the a_r are arbitrary coefficients. A retrieval of this type, which uses RSV as features instead of term weights, we call second level retrieval.

6.2 Goals for Batch Filtering

Two requirements must be met in order for a batch filtering system to perform well. It must be able to create a profile, which will generate a satisfactory retrieval. In the past TREC meetings there was a high correlation between the best retrievals and the best filtering scores. An additional challenge is to set the retrieval threshold to satisfy the target functions.

6.3 Methodology for Batch Filtering

Fig-1 describes a pictorial representation of the batch filtering procedure.

The FT92 Collection was indexed and statistics were collected by our standard PIRCS system. The collection was divided into two equal parts, a test collection and a training collection. The creation of the final filtered documents was a four-step process.

Step 1) Six retrieval profiles were created from the training subcollection using the Pircs system. They are listed below a run name abbreviation followed by a short description:

- (not1) pircs no training type 1
- (not2) pircs no training type 2
- (pircsb1) pircs type 1 expansion 250
- (pircsb2) pircs type 2 expansion 60
- (pircsf1) pircs type 1 expansion 40
- (pircsf2) pircs type 2 expansion 10

Step 2) Using the six profiles perform retrievals on the FT92 test subcollection. Combination coefficients are computed via a genetic algorithm based learning program. The GA attempts to maximize the average uninterpolated precision.

Step 3) The six profiles are recreated, using the full FT92 Collection. Of course the profiles not1 and not2 are unchanged since the do not learn from relevant documents. The profiles retrieve documents from FT92 and combined using the coefficients from step1. A logistic regression translates the retrieval status values into a probability.

Step 4) The six profiles are now applied to the FT93-94 collection. They are combined using the combination coefficients and transformed by the logistic regression coefficients. The values above the threshold are selected for filtering.

6.4 Selection of Filtering Threshold

There are two reasonable ways to select the cutoff point. One method is to calculate retrieval status value for which the Fi measure yields the maximum. If this occurs at multiple values select one of them. The other is to use logistic regression to transform the retrieval status value into a probability and use the probability for the cutoff. We used the first method prior to TREC7 (and in adaptive filtering) the second since then. Translating the retrieval status value into a probability is also very useful for the user of the system.

Only 43 topics had relevant training documents and we did not submit documents for the other 7. The quality of the training document may not be very good since they were selected by ad-hoc systems. The routing and filtering systems make use of the available judged documents to perform automatic term expansion and training, and consequently uncover more relevant documents. At the TREC7 conference 3301 relevant documents were found for the AP89 collection, while before TREC7 only 1598 were known. Thus the density of relevant documents was over twice as much as was assumed previously. Looking back at TREC7 we observed, that for our filtering run a .25 threshold we would do better at the average of .159 probability and a median of .07, and for the .40 threshold with an average of .298 and a median of .22! Consequently we decided to set the threshold at .30 probability for F1 and .15 for F2. Documents were selected for 30 topics for F1 and for 33 for F2.

The run names for batch filtering documents submitted are pirc9BF1 and pirc9BF2.

6.5 Batch Filtering Results

Subsequently we discovered that our submitted result contained some FT92 documents caused by an incorrect retrieval file. After deleting the FT92 documents from the filtered files, we recomputed the revised scores. Table 6.1 shows the official and revised results.

run	>	=	<
Pirc9BF1 official	19(7)	23(14)	8
Pirc9BF1 revised	25(10)	20(16)	5
Pirc9BF2 official	18(18)	29(15)	3
Pirc9BF2 revised	27(22)	21(16)	2

Table 6.1 Comparison of batch filtering results with median. Number in parenthesis is number of best values.

Compare levels	> ,= ,<	overall
F1		
.30:.40	12,12,8	+4
.30:.35	8,17,7	+1
.30:.25	12,10,10	+2
.30:.20	12,8,12	0
.30:.15	19,4,9	+10
F2		
.15:.25	17,6,9	+8
.15:.20	14,10,8	+6
.15:.12	13,10,9	+4

Table 6.2. Compare threshold levels for batch filtering.

Threshold	Score
BF1	
.30 official	295
.40	395
.35	395
.30 revised	399
.25	377
.20	415
.15	364
BF2	
.15 official	875
.25	746
.20	856
.15 revised	940
.12	964

Table 6.3. Score at threshold level

We also compared the performance of our revised runs to other threshold levels. Table 6.2 is a query by query comparison and Table 6.3 shows the total score for various levels. It is apparent from the tables that the decision to use lower levels was justified.

7. Routing Track

The focus of our routing runs is to experiment with our genetic algorithm combination of retrievals. The first run pirc9R1 combines 6 retrievals and the second pircs9R2 combines 8. The first submitted routing retrieval pirc9R1 was prepared the same way as the filtering retrieval. We created six profiles using the same expansion and training parameters as described earlier for filtering. They were combined using a GA attempting to maximize the average uninterpolated precision just as for filtering. We also used the same term statistics computed from the FT92 collection The difference is, that all the relevant documents from FT91 FT92 LA and FBIS were used for training.

For the pirc9R2, two more retrievals were added to the above six to generate the second submitted run. A pure ga based retrieval and a retrieval using backpropagation. For each topic 120 term were selected using our standard pircs system. To these we added 15 positive and 6 negative pairs. The ga optimizes the maximum likelihood measure, thus performing a logistic regression. The backpropagation neural network is a modified version of NevProp a publicly available c program maintained by Phil Goodman of the University of Nevada. No hidden nodes were used for the backpropagation training. In the past we did not have good results with these methods, but the diversity the produce usually enhances the combination.

Routing Retrieval Results

Run name	>	=	<
Pirc9R1	32(8)	8(2)	8
Pirc9R2	22(5)	8(2)	18

Table 7.1 Comparison of routing results with median. Number in parenthesis is number of best values.

The combination Pirc9R1 performed well. The ga and np retrieval did not, and adding it to Pircs9R1 depressed the result. We plan to investigate the cause of this. One possibility is that all evaluated documents were used for training, but the terms added to the query were only based on the relevant documents. These terms may have been underrepresented in the evaluated nonrelevant documents and thus their weight was inflated.

Table 7.2 compares the individual components to the combined retrievals. Max is the hypothetical retrieval that could be achieved if the best retrieval for each query

method	avg	% chg from not2	% chg from Pircsb1
not1	0.2182	23.1%	-47.3%
not2	0.1773	0.0%	-57.2%
Pircsb1	0.4008	126.1%	-3.2%
Pircsf1	0.4140	133.5%	0.0%
Pircsb2	0.3297	86.0%	-20.4%
Pircsf2	0.3273	84.6%	-20.9%
max	0.4670	163.4%	12.8%
Pirc9R1	0.4316	143.5%	4.3%
Pirc9R2	0.3990	125.0%	-3.6%

Table 7.2 Individual retrieval results.

	Pircsb1	pirc9R1	% chng
Rel_ret	1214	1203	-0.91%
at 0.00	0.7207	0.7380	2.40%
at 0.10	0.6463	0.6801	5.23%
at 0.20	0.5923	0.6245	5.44%
at 0.30	0.5410	0.5737	6.04%
at 0.40	0.5008	0.5132	2.48%
at 0.50	0.4425	0.4539	2.58%
at 0.60	0.3770	0.3941	4.54%
at 0.70	0.3096	0.3315	7.07%
at 0.80	0.2586	0.2679	3.60%
at 0.90	0.1816	0.1917	5.56%
at 1.00	0.1319	0.1425	8.04%
	0.4140	0.4316	4.25%
Precision:			
At 5 docs:	0.5080	0.4920	-3.15%
At 10 docs:	0.4020	0.428	6.47%
At 15 docs:	0.3533	0.376	6.43%
At 20 docs:	0.3190	0.328	2.82%
At 30 docs:	0.2740	0.2747	0.26%
At 100 docs:	0.1462	0.1516	3.69%
At 200 docs:	0.0940	0.0941	0.11%
At 500 docs:	0.0456	0.045	-1.32%
At 1000 docs:	0.0243	0.0241	-0.82%
Exact:	0.3985	0.4168	4.59%

Table 7.3 Routing Effectiveness Levels

was known. Pirc9R2 improved 4.3% over the best retrieval Pircsf1 but it did not reach the performance of max. Table 5.3 is a more detailed comparison of Pirc9R2 with the best performing individual retrieval Pircsf1.

8 Conclusion

In TREC8 experiments we coninued to demonstrate that our PIRCS system consistently return competitive results. For ad-hoc retrieval, multiple techniques such as combination of retrieval lists (data fusion), collection enrichment and 2-stage pseudo-feedback all can cooperatively boost effectiveness to the best level. For query track, we showed the importance of term choices in query formulation. For adaptive filtering track, we showed that minimally storing only selected documents can enable us to do filtering and adaptive threshold setting. Our utility scores are negative possibly due to difficulties in acquiring training data for this FT collection. Batch filtering and routing continues to do well.

References

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

HiKr99 Hiemstra, D & Kraajj, W. "Twenty-One at TREC-7: Ad-hoc and Cross-Language Track" In: NIST Special publication 500-242. pp.227-238, 1999.

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

KwCh98 Kwok, K.L. & Chan, M. "Improving Two-Stage Ad-Hoc Retrieval for Short Queries". In: Proc. 21st SIGIR'98Conference. pp.250-256, 1998.

KwGL95 Kwok, K.L, Grunfeld, L & Lewis, D.D. "TREC-3 Ad-hoc, Routing Retrieval and Thresholding Experiments using PIRCS" In: NIST Special Publication 500-225. pp.247-255.

Kwok95 Kwok, K.L. " A Network Approach to Probabilistic Information Retrieval". Journal ACM Transactions on Information Systems. Volume 13, pp.324-353 1995

VoHa99 Voorhees, E.M & Harman, D.K. Overview of the Seventh Text REtrieval Conference (TREC-7). In: NIST Special Publication 500-242. pp.1-23; 1999.

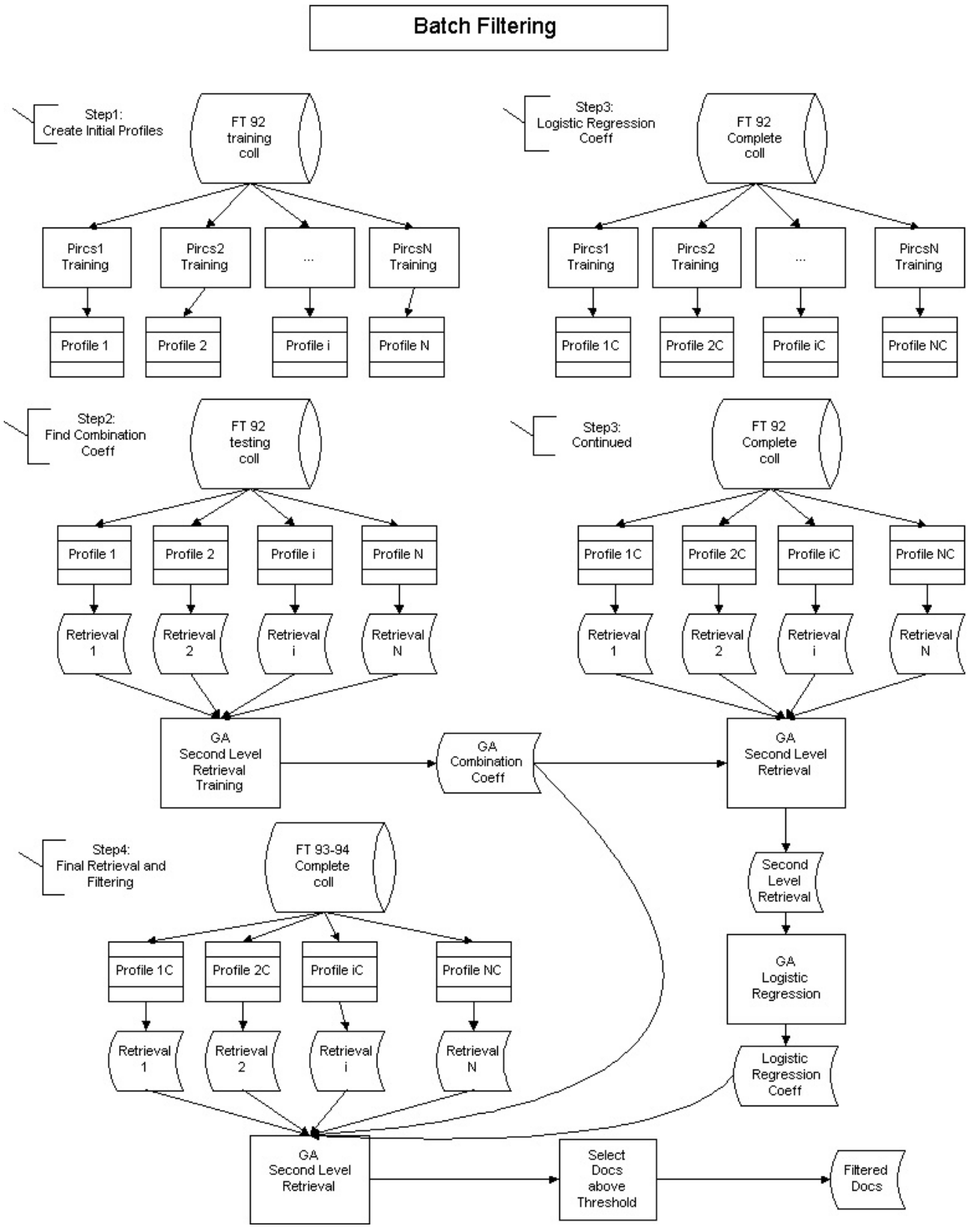

Fig.1: Batch Filtering System Flowchart