TREC-3 Ad-Hoc, Routing Retrieval and Thresholding Experiments using PIRCS

K.L. Kwok & L. Grunfeld
Computer Science Dept., Queens College, CUNY,
Flushing, NY 11367. email: kklqc@cunyvm.cuny.edu

D. D. Lewis
AT&T Bell Labs, 600 Mountain Avenue,
Murray Hill, NJ 07974. email: lewis@research.att.com

ABSTRACT

The PIRCS retrieval system has been upgraded in TREC-3 to handle the full English collections of 2 GB in an efficient manner. For ad-hoc retrieval, we use recurrent spreading of activation in our network to implement query learning and expansion based on the best-ranked subdocuments of an initial retrieval. We also augment our standard retrieval algorithm with a soft-Boolean component. For routing, we use learning from signal-rich short documents or subdocument segments. For the optional thresholding experiment, we tried two approaches to transforming retrieval status values (RSV's) so that they could be used to partition documents into retrieved and nonretrieved sets. The first method normalizes RSV's using a query self-retrieval score. The second, which requires training data, uses logistic regression to convert RSV's into estimates of probability of relevance. Overall, our results are highly competitive with those of other participants.

1. INTRODUCTION

PIRCS is an experimental information retrieval (IR) system designed and implemented at Queens College. PIRCS allows both ad-hoc retrieval and routing of free-text documents based on natural language as well as Boolean form queries. In TREC-1, Queens College took part in Category B, with PIRCS returning highly competitive results on a 1/2 GB test collection. A major redesign of PIRCS allowed it to be applied to the full 2 GB Category A data set for TREC-2. However, only a subset of PIRCS' capabilities had been reimplemented in time, leading to PIRCS results for TREC-2 that were median and below our expectations.

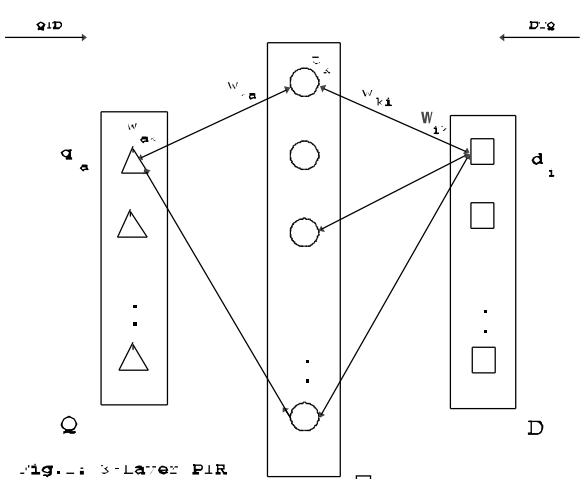
For TREC-3, we employ an upgraded version of PIRCS to the full TREC-3 ad-hoc retrieval and routing of English data sets, as well as taking part in the optional experiment on thresholding for routing. For ad-hoc, we make use of several best-ranked subdocuments for query enhancement to improve retrieval results. For routing, different subsets of relevant subdocuments are used for learning. Two approaches to thresholding were tried. The first approach normalizes the raw scores using the query self-retrieval value and applies a fixed threshold. The second estimates the probability of relevance for each test document using a logistic function fitted to the training data. We show two ways to use these probability estimates for thresholding.

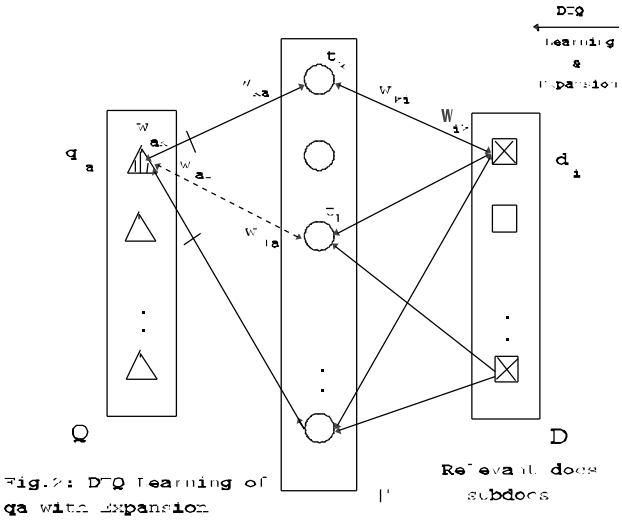
In Section 2, we review the salient features of PIRCS. Sections 3 and 4 discuss our ad-hoc and routing experiments respectively, while Section 5 is on thresholding. Our conclusion is in Section 6.

2. PIRCS' DESIGN

2.1 Effectiveness Aspects

PIRCS is based on the probabilistic indexing and retrieval models of [MaKu60, RoSp76] but extended with the concept of document components [Kwok90]. Both documents and queries are represented as lists of independent conceptual components that are approximated with content terms. These terms can be single words or two-word phrases and are obtained automatically from the corpus. Additionaly, queries can also have a Boolean expression structure. Term weights are based on conditional probabilities estimated from local term usage in individual items (documents or queries) and from global term usage in the whole collection. Thus, within-item term frequencies, item length, collection and inverse collection term frequencies (ICTF) and total number of tokens are effectively utilized. Moreover, PIRCS can bootstrap itself (i.e. define initial weights) based on the property that each individual item is consisted of conceptual components that are relevant to the item. The probability estimates relying





on item self-relevancy are quite rough because the sample of components from a single item is small at this initial stage, but may become more accurate when the system learns from larger relevant samples obtained via historical or interactive relevance judgment.

The system is implemented as a network shown in Fig.1, where the three layers of nodes (Q, T, D) denote the set of queries, terms and documents. Nodes of one layer are connected to those of an adjacent layer only, via bi-directional edges with the weights discussed earlier. In a heterogenous collection one has to deal with documents of widely different lengths, or that one document may contain several unrelated stories. We segment these documents into 'subdocuments' of more uniform sizes as before, and nodes on our network can be subdocuments or whole documents. This segmentation may lead to better retrieval, display, learning, and general efficiency in document processing as discussed in [KwGr94a,b].

During retrieval, PIRCS normally supports two modes: query-focused, when activation starts from a document and spreads towards a query under attention, or vice versa for the document-focused mode. This results in two retrieval lists for each query with different RSV's (retrieval status values) that are based on different statistics. When these RSV's are additively combined, the resultant ranking of documents almost always provides better effectiveness than either one alone. In addition, a third retrieval list can be provided if a Boolean expression tree replaces each of the set of query-term edges of the network. Thus, three types of evidence can be combined to enhance retrieval results.

An important feature of our network is that it supports training based on given relevant items. Thus, if a query has a known set of relevant documents as shown in Fig.2 (and vice versa, a document having a known set of relevant queries), the edge weights on either or both sides of the net can adapt incrementally based on term usage statistics in the relevants and attain values that would provide more effective retrieval results. Moreover, the network also supports query expansion whereby terms that are used in relevant items are selectively added with appropriate weights to the query under attention. Thus, these procedures can lead to higher recall because relevant documents having terms not contained in the original query may get retrieved, and higher precision because terms that are prevalent in relevant documents would attain larger weights and documents having them may get ranked earlier. With these learning capabilities, the network behaves like a 2-layer neural network with adaptive architecture [Kwok9x].

2.2 Efficiency Aspects

PIRCS is also unique in its processing design [KwGr94]. In our system a full inverted (posting) file is not produced as is normally done in many other systems, saving substantial disk space and reducing the 'dead time' between a collection being acquired and its availability for searching. Also, the problem of updating the full inverted file when a collection grows incrementally becomes non-existent. We create our network for learning and retrieval dynamically at query time from the direct file consisting of the list of terms within each document. The network is built based on the 'active term set' defined by the current queries and subcollection, ensuring that it is compact enough to fit into main memory. Having a network image of a textbase in RAM allows us to do learning and retrieval relatively fast.

Since TREC-2, we have reduced the time for building the network by over 80% by truncating and compressing the direct file. Depending on the query length, a single TREC topic can now be run against a 1/2 GB textbase in about 2 minutes clock time on a Sparc 10/30, producing a ranked list of 1000 subdocuments. Batching queries 10 at a time reduces the average to slightly more than one minute per query. Although this is not yet satisfactory for interactive retrieval, it is quite adequate for experimentation. We expect future upgrades will improve on this timing. For a raw text collection of x GB, we can support retrieval using only about 1.2x GB with our truncated and compressed direct file that does not include term position information. During preparation of the textbase however, more space is needed.

PIRCS also employs a master-subcollection file organization in such a way that a huge collection can be broken up into convenient subcollection sizes of say 0.5-1.0 GB each, while ensuring that documents are ranked exactly as if a single large collection and its statistics had been used. This gives us the flexibility to organize, retrieve and maintain collections of essentially unlimited sizes bounded only by available hardware. If PIRCS runs in a parallel environment with multiple processors each supporting one subcollection, then timing would also not be much more than that for a single subcollection. In addition, the previously mentioned subdocument segmentation approach to handling heterogenous documents of unconstrained lengths also contributes to the efficiency of processing.

3. AD-HOC RETRIEVAL EXPERIMENTS

3.1 Methodology

Ad-hoc experiments for TREC-3 involve creating a textbase from the revised Disks 1&2 data and retrieving with the new, unseen 4th topic set (151-200). We segment the documents into about 550-word chunks ending on a paragraph boundary. Four subcollections were produced, viz.: wa1 (274,110), fzd1 (422,660), wa2 (239,107) and fz2 (149,790) totalling 1,085,667 subdocuments. They are served by a master lexicon of 658,404 unique terms that includes 55,296 entries of our semi-automatic 2-word phrases. Our stop list remains as in TREC-2 with 630 words. The title, description and narrative sections of each topic are processed automatically like documents (but with non-content introductory phrases and 'not' sentences removed), resulting in query set Q4. A Boolean expression for each topic is also automatically generated by ORing two AND clauses that are constituted of the content terms in the title and description sections. A new feature in TREC-3 is that the concept section previously supplied with each topic is no longer available, making retrieval more difficult.

Our two submitted results are pircs1 (evaluated by NIST) and pircs2. An unsubmitted run called pircs0 consisting of our standard combination of query-focused and document-focused retrieval procedures (as discussed in Section 2) serves as a control for our internal comparison. In pircs1, we augment pircs0 by a new document-focused activation spreading which we called 'controlled recurrent processing'. In this process we let the system do some learning and query expansion from a few of the best-ranked sub-documents of each query. Ad-hoc is an initial retrieval and therefore no known relevant documents are available for training. However, previous experiments [KwPK93, KwGr94] have shown that PIRCS' ad-hoc operation can achieve average precision close to 60% in the first 5 to 10 retrieved. Hence such feedback without user judgments could be useful, and some of our limited experiments showed that it was indeed the case. It is fairly simple to implement the process in our network (Fig.1). Activation first spreads from a query node to the term layer and then to the document layer nodes. That would be our normal document-focused process. Instead, we initiate a choice operation that selects the m best-ranked document nodes with sufficient activation, and let them spread backwards to the term layer thus turning on many terms not originally connected to the query. Again, only the n highest activated terms are selected, and activation then spreads back and impacts on the document nodes providing the final RSVs for ranking. The process is very efficient; timing is practically unchanged from not using the controlled recurrent process. Moreover, it is done 'on-the-fly' so that future additions to the subcollection will influence the outcome in a dynamic fashion. We use a conservative setting of m=6 and n=30. The result is that only 40 out of 50 queries are modified, and for these an average of about 11 (443/40) new terms are added per query.

For our second ad-hoc submission pircs2, we augment our normal pircs0 with soft-Boolean retrieval. For each query a Boolean expression consisting of the OR of two AND clauses is automatically generated. Each of the AND clauses is composed of the content terms from the <topic> and <desc> sections of a topic respectively.

3.2 Results

A summary comparison of our approaches to ad-hoc retrieval averaged over the 50 queries in Q4 is tabulated in Table 1 below. Compared with our standard approach pircs0, both of our enhancements in pircs1 and pircs2 are successful. In particular, the use of a few best-ranked subdocuments for query training and enhancement (pircs1) improves over pircs0 by 8.6% in average precision, 6.6% in R-Precision and 0.6% in relevants retrieved (rel_ret). The effect on recall is negligible. Adding soft-Boolean processing is also effective in contrast to TREC-2, leading

	pircs0 (std)	pircs1 (Qexp)	pircs2 (Bool)
relev:	9805	9805	9805
rel_ret:	5981 (base)	6017 (+0.6)	6145 (+2.7)
av.precision:	.2764	.3001 (+8.6)	.2913 (+5.4)
precision at:			
5 docs	.6520	.6600	.6600
20 docs	.5090	.5670	.5240
100 docs	.3540	.3792	.3774
R-precision:	.3267	.3484 (+6.6)	.3402 (+4.1)

Table 1: Ad-Hoc Results

to increases of 5.4% in average precision, 4.1% in R-precision and 2.7% in rel-ret. We believe that the absence of the concept section in the topics (which normally has many specific keywords) has depressed the standard pircs0 result by something like 20 to 30%. This offers an opportunity for the controlled recurrent processing and the soft-Boolean operations to show some improvements.

Using pircs1 results, we observe that it recalls 61.4% of all relevants at 1000 documents retrieved. At 5 documents retrieved, one can expect more than 3 of them are relevant; at 20 documents, more than 11 are relevant; and at 30 more than half.

Comparisons with the MEDIAN values of submissions from other TREC-3 sites are summarized below in Table 2. It can be seen that both pircs1 and pircs2 results outperform the median. Since different sites make use of different techniques and methodologies, it would be interesting to see for each query which site has done best as tabulated in [Harm94] and why. We also introduce in TREC-2 MAXI-retrieval as a hypothetical system that returns the best performance for each query among all sites. This assumes that we have an intelligent agent who is able to choose the best retrieval system among this set of participants for each query, and would reflect the best we can do using our

	pircs1			pircs2		
	>	=	<	>	=	<
av. prec:	31(1)	2	17	36(0)	1	13
rel_ret @ 100:	27(1)	3	20	30(1)	6	14
rel_ret @ 1000:	30(2)	2	18	34(1)	3	13

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

Table 2: Comparison of Ad-Hoc Results with Median

collective wisdom at this time. This MAXI-system will return an average precision, precision at 100 docs and at 1000 docs of 0.4982, 0.5736 and 0.1600 respectively. Thus, Pircs1 achieves 60.2%, 66.1% and 75.2% and pircs2 achieves 58.5%, 65.8% and 76.8% respectively of these best values. Pircs1 seems to be better than pircs2 in absolute values of precision, but pircs2 outperforms pircs1 when we count the number of query results that are above median.

4. ROUTING RETRIEVAL EXPERIMENTS

4.1 Methodology

Routing involves the 3rd topic set (101-150) processing against Disk3 data. Our routing queries Q3 are defined using both the topic texts as well as their known sets of relevant subdocuments from Disks 1&2. No Disk 3 data were used for training as no relevant items from Disk 3 were available. As discussed before [KwGr94a,b], a judicious choice of training items can impact favorably on retrieval results. Short documents are the quality items and they are very effective for training. Selecting only them for training is also efficient because there is no need to do a ranking operation. However, there may not be enough of them in the known relevant set. Selecting the top-ranked subdocument of every known relevant ensures that every item is represented for training, and these subdocuments can be the most signal-rich portions of the relevants. This is an advantage of segmentation because using the top whole document may add noise in the learning process if the document happens to be long and most of it is irrelevant. However, a ranking operation is required. In [KwGr94b] we also show that the usual feedback strategy of using the x best-ranked items for training can lead to inferior retrieval results.

For TREC-3, we segment long documents and documents with multiple unrelated stories as previously, but use a larger chunk size of 550 words instead of 360 [KwPK93, KwGr94]. This saves some space by reducing the number of our network nodes. Also, if we get comparably good results as in our previous TREC-2 'further' experiments, it can be argued that segmentation size is not critical. We submitted two sets of results based on the two selection strategies discussed earlier. Pircs4 makes use only of the short relevant documents for each query, i.e. those having 160 or less unique words per document (after deleting stop words and conflating via stemming). They roughly correspond to those of TREC-2 'nonbreak' documents of about 360 raw words, and are independent of how we segment documents. There are 4,043 such short documents out of a total of 11,645 relevants. These relevants break up into 32,837 subdocuments. Thus, for pircs4 we make use of about 12% of the relevant texts only. In pircs3 which is

our official routing submission evaluated by NIST, we further merge this set of short training documents with the set of top-ranked relevant subdocuments obtained from 2,400 best-ranked subdocuments for each query. ('Top-ranked' means the set of subdocuments, only one from each document, that are ranked highest, while 'best-ranked' just means the highest ranked set of subdocuments). The 2,400 best-ranked's may not be all relevant and may have many subdocuments from the same document; so the top-ranked relevants obtained are less than 2,400 and we do not recover all 11,645. After merging with the short documents we end up with 11,559 subdocuments for training purposes. This represents about 35% of the available relevant texts. The actual process of training on our network is unchanged from TREC-2.

4.2 Results

Comparison of our two approaches for routing (see the appendix of this volume) shows that training by short relevant documents only (pircs4) is both effective and efficient, as observed previously [KwGr94a]. It gives an average precision of 0.3749, R-precision of 0.3899 and rel-ret at 1000 documents of 7318 (78.24% of all relevants). Augmenting the training document set by using a ranking operation to add top-ranked subdocuments (pircs3) involves much more work, and leads to average precision of 0.3887, R-precision of 0.3991 and rel_ret of 7640 (81.69% of all relevants). These are 3.7%, 2.4% and 4.4% better tban pircs4. If an application does not require the last few percent improvements in precision and recall, training with short documents will do. The precision at 15 documents or less retrieved is at least 60% on average. This means that one would expect 9 relevant documents in the first 15 retrieved averaged over the Q3 (101-150) query set; and in the first 5 retrieved, over 3.

Comparison with the MEDIAN values of submissions from other TREC-3 sites are summarized below in Table 3. It can be seen that both pircs3 and pircs4 results are substantially better than median. The MAXI-system returns

	pircs3			pircs4		
	>	=	<	>	=	<
av. prec:	41(4)	1	8	36(1)	0	14
rel_ret @ 100:	36(6)	8(1)	6	34(5)	5	11
rel_ret @ 1000:	38(10)	9(5)	3	33(7)	12(5)	5

(figure in paranthesis is number of queries equaling the values; some median values are also the best values)

Table 3: Comparison of Routing Results with Median

an average percision, precision at 100 and at 1000 documents of 0.4951, 0.5342 and 0.1631 respectively. Pircs3 achieves 78.5%, 86.2% and 93.7% and pircs4 achieves 75.7%, 84.4% and 89.8% respectively of these best results. This indicates that pircs performs comparatively well for exhaustive searches at the high recall end. Also pircs3 returns 7640 relevants retrieved at 1000 documents cutoff, the most of the automatic systems.

Tabulated below in Table 4 is the behavior of our routing algorithm at different levels of query expansion, and can be directly compared with our pircs3 results at expansion level 80. Effectiveness of our retrieval algorithm seems to plateau at the high end of the expansion. The behavior is quite similar to what we observed in our TREC-2 'further' experiments. We did not perform massive query expansion as done in [BuAS94].

Expansion Level:	0	40	80	120
relev:	9353	9353	9353	9353
rel_ret:	7015	7574	7640	7680
average precision:	.3256	.3782	.3887	.3960
Precision at:				
5 docs	.5360	.6480	.6400	.6560
10 docs	.5200	.6080	.6260	.6480
100 docs	.3844	.4490	.4606	.4698
R-Precision:	.3668	.3862	.3991	.4102

Table 4: Pircs3 Routing Results at Various Expansion Levels

5. THRESHOLDING EXPERIMENTS FOR ROUTING RETRIEVAL

Routing systems are presented with new documents in an ongoing fashion. Limited storage and/or limited communication bandwidth may force the system to discard all but the most relevant documents. The system must not only rank documents, but must strictly partition them into retrieved and nonretrieved sets, just as set-based retrieval (boolean query) systems and text categorization systems do.

For systems which produce RSV's, one approach to partitioning is to set a threshold and retrieve all documents with RSV's over that threshold. Thus the TREC-3 optional routing experiment requiring systems to declare which documents were retrieved was refered to as a ''thresholding'' experiment. It should be noted, however, that thresholding is not the only approach to partitioning [Jaco94,ToAp94].

In this section we begin by discussing some problems with the TREC-3 thresholding evaluation. We then present our approach to the evaluation, our results, and some observations.

5.1 Evaluation

A person setting up a routing system needs to know what users desire the routing system to do. This goal needs to be expressed in the form of some effectiveness measure which the system can be tuned to optimize. The goal for the TREC-3 thresholding experiment was defined by NIST as follows:

> "This threshold should be 'set' so that the system would retrieve R relevant documents, where R is the number of relevant documents that were found for this topic in TREC-2."

Retrieving a specified number of relevant documents is one of several plausible goals for a routing system. As defined above, however, the goal is unclear. Only the ideal partition of the test data (R relevant documents retrieved and, presumedly, no nonrelevant documents retrieved) is defined. It is not clear how non-ideal partitions should be evaluated. For instance, is retrieving 50% of the relevant documents and no nonrelevant ones better or worse than retrieving 90% of relevant documents plus 10% of the nonrelevant ones?

In any case, the thresholding experiment's goal was incompatible with the overall rules of the TREC-3 routing evaluation:

> "Special care should be used in handling the routing topics. In a true routing situation, a single document would be indexed and 'passed' against the routing queries. Since most of you will be indexing the test data set as a complete set, routing should be simulated by not using any test document collection information (such as IDF based on the test collection, total frequency based on the test collection, etc.) in the searching."

This meant, in particular, that routing methods were not allowed to take into account the number of documents in the test set or other global properties of the test set. But without such knowledge, it is impossible to define a rule that is expected to retrieve R relevant documents. To see this, consider a rule that retrieves R relevant documents on test set A. Now consider test set B which consists of 100 copies of the test set A. Clearly 100*R documents will be retrieved on the test set B, and under the TREC-3 rules there would be no way to distinguish the two test sets and thus, in general, no way to meet the goal of the thresholding experiment given the TREC-3 rules. (It is worth noting that there are effectiveness measures, notably decision theoretic loss measures [DuHa73], which could have been optimized under the TREC-3 restrictions.)

5.2 Approach

Despite the problems with the thresholding evaluation, we chose to test several approaches to it. A common goal of these approaches was to normalize the PIRCS RSV's so that a threshold could be determined automatically for each topic. In PIRCS, the retrieval status value that is calculated for each document is theoretically the log-odds value that the given document is relevant to the topic under attention. In practice, so many approximations are made that the raw RSV's are only useful for ranking.

5.2.1 Normalization Using Query Self-Relevance

Our first approach to normalizing RSV's relies on the property of 'self-relevance' as discussed in [Kwok85]. A query describes what a user needs. Suppose there were a 'document' identical to the query. We can presume this hypothetical document would be relevant to the query. Thus, given a query we always have one 'relevant document' as reference. We call the RSV for this 'document' the query self-retrieval RSV, or qRSV.

The qRSV for a query will normally be larger than the RSV for any true document, since true documents will only partially match the query. If we divide a document's RSV by qRSV, the result can be interpreted as a percentage of the RSV of a good 'relevant document'.

In practice, we use (constant * RSV/qRSV) as the normalized score. The hope is that this normalized score is more comparable between queries, so that to achieve a desired effect the same threshold can be used for all queries. For the TREC-3 thresholding experiment all test documents with normalized scores of 0.5 or above were considered to be retrieved. We call this strategy Self1. The threshold of 0.5 was chosen arbitrarily, and could be improved by learning from other queries, or from relevance judgments on a particular query. A related strategy, based on the score of the top ranked document rather than a self-retrieval value, has been used elsewhere [LPY94].

5.2.2 Normalization Using Logistic Regression

For our other approaches, the normalized RSV was an explicit estimate of the probability of relevance. We began by finding, for each topic, the RSV for all training documents which were judged for relevance to that topic. The resulting RSV's, along with the relevance data, were fit

to a logistic function. The result was a pair of parameters a and b for each topic, such that

$$\frac{\exp(a + b * RSV)}{1 + \exp(a + b * RSV)}$$

(where exp is the exponential function) should be a good estimate of P(T|D), the probability that document D is relevant to the topic [LeGa94]. The routing system can then generate an estimate of P(T|D) for new documents by first computing an RSV for the document and then applying the above formula with the appropriate a and b for that topic. Note that unlike Self1, this method requires relevance judgments.

Our submitted thresholding run, Prob1, was a compromise between the stated goal of the threshold evaluation and the somewhat contradictory restrictions on how routing could be done. We estimated P(T|D) for all training documents for a topic and sorted these estimates in descending order. A program then went down this sorted list, taking the cumulative sum of the P(T|D) values until the sum exceeded R (the number of documents judged relevant to the topic). The value of P(T|D) at that point was chosen as the threshold to be used on the test set. (Actually, this value was used in a normalization formula for RSV's, enabling 0.5 to be used as the threshold for all topics, but the effect was the same.) The rationale was that the summed P(T|D)'s provided a better estimate of the number of relevant above a threshold, than the number of judged relevant documents above a threshold.

As observed earlier, however, there was no reason to believe that retrieving a specified number of relevant documents was possible at all without knowledge of the size of the test set and the distribution of scores on it. We therefore tested a third method (Batch1). In this approach, the RSV's of test documents instead of training documents are converted to estimates of P(T|D), using the logistic parameters fit on the training data. (No information about test set relevance judgments is used.) Again a program sorts these estimates and sums them in descending order until a total of R is reached. Exactly the documents above that point were retrieved.

Note that Batch1 is not allowed under the TREC-3 rules, since all test documents must be scored and sorted before the decision is made to retrieve any individual document.

5.3 Results

Table 5 below shows for all three partitioning methods, the total number of documents retrieved (i.e. above threshold), estimated number of relevant retrieved (for methods that can estimate this), and the judged number of relevant retrieved, along with macroaveraged recall, precision, and R-precision as computed by the TREC-3 thresholding evaluation program.

	Total Ret	Estimated Rel Ret	Judged Rel Ret	Recall	Prec.	R-Prec.
Self1	23872	---	6708	0.7118	0.2658	0.2795
Prob1	10498	---	4562	0.5145	0.3916	0.4477
Batch1	43499	10155	7014	0.8362	0.1907	0.1765

Table 5: Thresholding Results

The desired number of relevant to retrieve (based on the training set) was 11,645. Batch1 comes closest to this target, with Self1, due to a fortuitous choice of threshold, close behind. Prob1, the submitted result, was inferior to the other two on this measure.

Batch1 is more effective than the immediate comparison of 7,014 relevant retrieved to 11,645 desired suggests. First, there were only 9,353 topic / test document pairs judged relevant at all, so this was the maximum number any system could get. Second, following the TREC-3 rules, we did not allow Batch1 to retrieve more than 1000 test documents for any topic, nor did we allow it to compensate for this limitation by attempting to retrieve more than the desired number of documents for other topics.

These limitations are reflected in the fact that the sum of the estimated probability of relevance for the 43,499 documents retrieved by Batch1 was approximately 10,155, which was lower than the target number of 11,645. Self1 was also affected by the restriction to 1000 documents retrieved since the 1000th document was still above the Self1 threshold for some topics.

Finally, not all documents which Batch1 retrieved had been judged for relevance. Of the 43,499 documents that Batch1 retrieved, only 29,193 were judged for relevance. Of those, Batch1 estimated that 8,037 were relevant, only a 14.6% overestimate compared with the true value of 7,014 relevant among these documents. Batch1 retrieved 14,306 documents which were not judged for relevance, and estimates that 2,118 of these are in fact relevant. Thus the effectiveness of Batch1, as well as of all the thresholding methods, is understated.

Batch1's 15% overestimation is similar to that observed by Cooper [CCG94] when applying a different logistic model to the TREC-2 data. A plausible explanation is that the training data, since it consists of best-ranked documents from a number of systems, contains more relevant

documents than a random sample with the same distribution of RSV's.

If we evaluate by R-precision, say, instead of by the stated thresholding goal of number of relevant documents retrieved, Batch1 is clearly inferior to both Self1 and Prob1. This reinforces our point that a meaningful evaluation of partitioning systems must define an actual effectiveness measure, not simply an ideal retrieval.

As a final obsevation, note that Batch1 estimates there are 2,118 unjudged relevant topic / document pairs among its retrieved data. Even allowing for the roughly 15% overestimation observed on the judged data, this would still suggest there are 1,840 unjudged relevant documents in just the Batch1 retrieved sets. This should be taken into account in interpreting the TREC-3 results.

6. CONCLUSION

The PIRCS retrieval system and its learning capabilities have been demonstrated to give excellent performance in a routing environment. In particular, we have shown that learning from known relevants can be beneficially affected by using only the signal-rich short documents and subdocuments for training and query expansion. For ad-hoc, the controlled recurrent activation spreading procedure, which enhances retrieval via a few initially best-ranked documents, appears useful. However, more investigation needs to be done to find the best parameters for this operation. For thresholding, the normalization of retrieval raw scores by the query self-retrieval value also seems to give reasonable results. An advantage of such an approach is that it needs not rely on past results for training (although it can make use of training data as well), and can be used in an ad-hoc environment for stopping retrieval. When training data is available, transformation to probabilities is useful for estimating and optimizing various effectiveness measures.

REFERENCES

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

[CCG94] Cooper, W.S., Chen, A. & Gey, F.C. Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. In: The Second Text Retrieval Conference (TREC-2). Harman, D.K. (Ed.). NIST Special Publication 500-215, 1994, pp. 57-65.

[DuHa73] Duda, R.O. & Hart, P.E. (1973). Pattern Classification and Scene Analysis. New York, Wiley-Interscience.

[Harm94] Harman, D.K. Overview of the Second Text REtrieval Conference (TREC-2). In: The Second Text REtrieval Conference (TREC-2). Harman, D.K. (Ed.). NIST Special Publication 500-215, 1994, pp. 1-20.

[Jaco94] Jacobs, P.S. GE in TREC-2: Results of a boolean approximation method for routing and retrieval. In: The Second Text Retrieval Conference (TREC-2). Harman, D.K. (Ed.). NIST Special Publication 500-215, 1994, pp. 191-199.

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

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

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

[Kwok85] Kwok,K.L (1985). A probabilistic theory of indexing and similarity measure based on cited and citing documents. J. of American Society for Information Science, 36(5):342-351.

[Kwok9x] Kwok, K.L (199x). A network approach to probabilistic information retrieval. ACM Transactions on Office Information Systems, to appear.

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

[LeGa94] Lewis, D.D. & Gale, W.A. A sequential algorithm for training text classifiers. In: SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Croft, W.B. and van Rijsbergen, C.J. (Eds.). Springer-Verlag, London, 1994, pp. 3-12.

[LPY94] Liddy, E.D., Park, W. and Yu, E.S. (1994). Text categorization for multiple users based on semantic features

from a machine-readable dictionary. ACM Transactions on Office Information Systems, 12(3):278-295.

[MaKu60] Maron M.E & Kuhns, L.J (1960). On relevance, probabilistic indexing and information retrieval. J. ACM 7:216-244.

[RoSp76] Robertson, S.E. & Sparck Jones, K. (1976). Relevance weighting of search terms. J. of American Society for Information Science, 27:129-146.

[ToAp94] Tong, R.M. & Appelbaum, L.A. Machine learning for knowledge-based document routing (a report on the TREC-2 experiment). In: The Second Text Retrieval Conference (TREC-2). Harman, D.K. (Ed.). NIST Special Publication 500-215, 1994, pp. 253-264.