Learning from Relevant Documents in Large Scale Routing Retrieval

K.L. Kwok and L. Grunfeld

Computer Science Department
Queens College, City University of New York
Flushing, NY 11367

ABSTRACT

The normal practice of selecting relevant documents for training routing queries is to either use all relevants or the 'best n' of them after a (retrieval) ranking operation with respect to each query. Using all relevants can introduce noise and ambiguities in training because documents can be long with many irrelevant portions. Using only the 'best n' risks leaving out documents that do not resemble a query. Based on a method of segmenting documents into more uniform size subdocuments, a better approach is to use the top ranked subdocument of every relevant. An alternative selection strategy is based on document properties without ranking. We found experimentally that short relevant documents are the quality items for training. Beginning portions of longer relevants are also useful. Using both types provides a strategy that is effective and efficient.

1. INTRODUCTION

In ad hoc Information Retrieval (IR) one employs a user-supplied free-text query as a clue to match against a textbase and rank documents for retrieval. In a routing environment, one has the additional option to consult a user need's history to obtain a set of previously judged documents. This set may be used with an automatic learning algorithm to help refine or augment the user-supplied free-text query, or even to define the query without the user description. We focus on employing the judged relevant set in this paper. (Judged nonrelevant documents have not been found to be useful in our model.) For this option, one needs to consider two separate processes:

(1) selecting the appropriate relevant documents or portions of them for training; and

(2) selecting the appropriate terms from these documents, expand the query and then effectively weighting these terms for the query.

It is well-known from TREC and other experiments [1,2,3,4,5,6,7,8] that process (2) can improve routing results substantially. However, process (1) is normally not given much consideration. One either uses all the relevant documents, or employs the best n of them after ranking with respect to the query under consideration. However, over time in a large scale environment, hundreds and thousands of such relevant documents may accumulate for each user need. A strategy of which and what parts of the relevant documents are to be employed for training needs to be considered. Would portions of relevant documents be sufficient? One reason for using a portion is that many documents can be long and may contain extraneous paragraphs and sections that are irrelevant. Using them for learning may contribute ambiguities during the term selection, query expansion and weighting processes. The problem is that current relevance information gathering is for whole documents only, and not at a more specific level such as which sentence or paragraph that is relevant. This problem would be alleviated if users are diligent and indicate the relevant components of a document that are actually relevant. However, this could be a burden that some users may want to avoid. It is therefore useful to have an algorithm to locate the most useful relevant components for training purposes. Another reason to use only portions of the relevants is consideration of efficiency: one would like to avoid processing long documents when most of it is irrelevant, or decrease the number of documents to be processed. This investigation concerns exploring ways to effectively choose a subset of documents for training a given set of routing queries.

2. PIRCS RETRIEVAL SYSTEM

PIRCS (acronym for Probabilistic Indexing and Retrieval -Components- System) is a network-based system implementing a Bayesian decision approach to

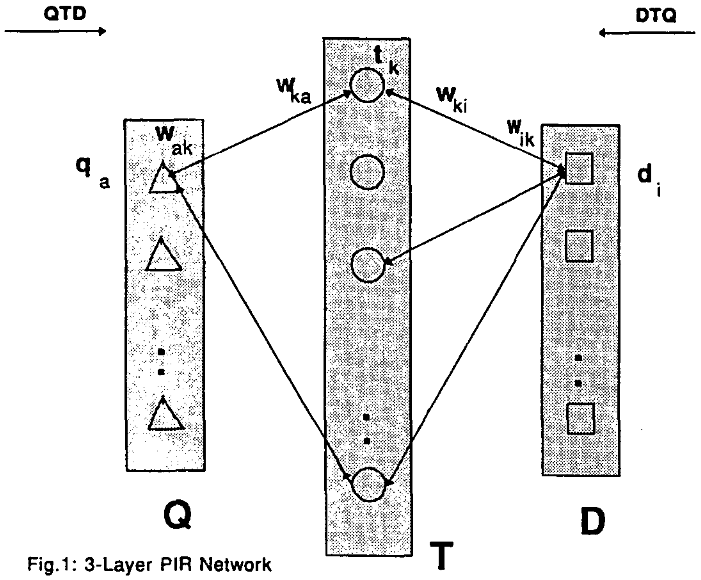

Fig.1: 3-Layer PIR Network

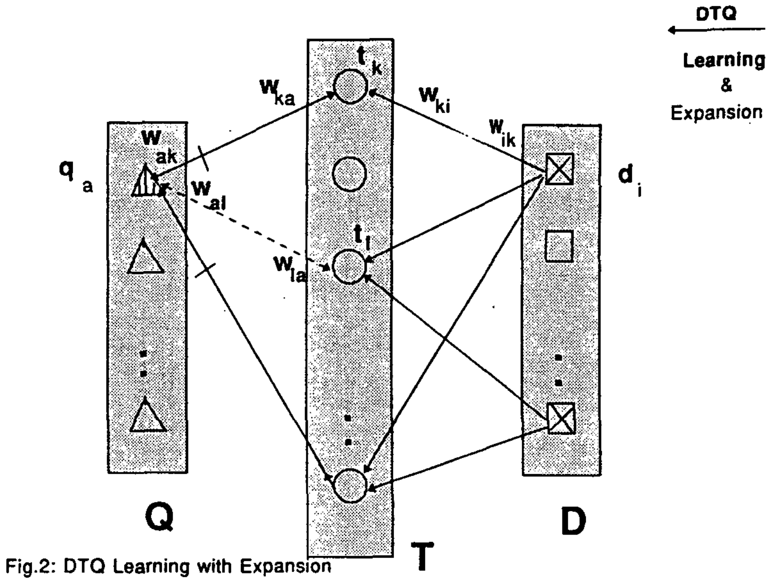

Fig.2: DTQ Learning with Expansion

IR [9,10] and extended with the concept of document components [11] as shown in Fig.1. The network [12] has three layers of nodes representing the queries (Q), terms (T) and documents (D), with edges connecting adjacent layers in a bidirectional fashion. Retrieval operation consists of initializing a document node d_i to activation 1 and spreading it via the edge weights to terms t_k and to a query node q_a under focus. q_a receives activation $\Sigma_k w_{ak} w_{ki}$ which is regarded as the query-focused retrieval status value (RSV) of d_i for ranking purposes. If activation originates from a query q_a and spreads towards d_i we accumulate the document-focused RSV: $\Sigma_k w_{ik} w_{ka}$ that is based on statistics of term usage different from before. Combining the two can cooperatively provide more effective results.

The edge weights of the net are first initialized with default values using global and local term usage statistics. Later they can learn from experience as illustrated in Fig.2. In particular for routing experiments, the edges on the query-term side of the net is first created based on the routing queries and the terms of the training collection, and given default values called self-learn relevant weights. Relevant training documents are then linked in on the document-term side of the net. Knowing which document is relevant to which query allows edge weights on the term-query side like w_{ak} to adapt according to the term usage statistics of the relevant sets via a learning rule that is borrowed from artificial neural network studies. New edges like w_{al}, w_{la} can also grow between queries and terms using, for example, the K highest activated terms of the relevant documents, a process we call level K query expansion. After learning, these query-term edges and weights are frozen, the training documents removed, and new unseen testing documents are then linked in for simulation of the routing operation. Thus, test documents are ranked with respect to each routing query based on term usage statistics seen in the training collection and the relevant documents.

3. RELEVANT SUBDOCUMENT SELECTION STRATEGIES

Our approach to uneven full text collections [3,6,8] has been to segment long documents on the next paragraph boundary after a run of 360 words, giving more uniform length subdocument units. Documents with unrelated multiple stories with detectable separation markers are also segmented at the markers. This approach may impact favorably on: 1) precision because shorter, more local units may diminish chance occurrence of terms used in senses different from what is intended; 2) term weighting because unrealistic probability estimates of term weights may be avoided; 3) query training and expansion because long documents may have unrelated and irrelevant topics and concepts that can add noise to these operations; 4) retrieval output display because one can narrow down to the relevant portion of a long document for the user; and 5) general efficiency because of handling multiple, more uniform subdocuments instead of one long document. In the TREC collections, documents of thousands of words long are not uncommon, and an example of a really long document is in the Disk1 Federal Register: FR89119-0111 with 400,748 words. With respect to item 3) query training and expansion, having many of these long documents in the training set would not only overwhelm our system but also lead to ambiguity and imprecision. Segmenting them into subdocuments may provide us with strategies in selecting the appropriate relevant portions of documents for learning. In the next subsections we consider document selection methods that can be broadly classified into three types: approaches based on document properties only, approaches based on ranking, and on combinations of both.

3.1 Subdocument Selection Based on Document Properties

These selection methods employ some heuristics on the properties of documents. Because they are based solely on a list of known relevant subdocuments they can bring in concepts that are not explicitely stated or related to the query. These methods are also efficient because no ranking operation is required. A risk of this type of approach is that if the selection method is not well designed, many irrelevant portions of relevant documents may be included for training and becomes counter-productive. Four methods have been experimented with and the rationale for their choice are given below:

(a) Use all subdocuments for learning and query expansion. This is the usual approach in small collections. In a large scale environment it may have the drawback of ambiguity, imprecison and inefficiency discussed in Section 1, but will serve as a basis for comparison.

(b) Use only relevant documents that 'break' into a maximum of max subdocuments. This effectively means eliminating long documents for learning, and may diminish ambiguities that come with them. Short documents should be more concentrated and focused in their content, and can be considered as quality items for training. In particular, max=1 means employing only 'nonbreak' documents. This was the strategy used in the original submitted results of our TREC-2 experiments. However, if the given relevants are mostly long, we may artificially diminish the available number of relevants used for training.

(c) Many articles including scholarly documents, certain newspaper and magazine items introduce their themes by stating the most important concepts and contents at the beginning of a document. They also summarize at the end. Therefore another approach is to use only the first or last subdocuments for training. Because of the way we segment documents so that some last subdocuments may be only a few words long, and the fact that some Wall Street Journal articles can have multiple unrelated stories within a document, we can only approximate our intent with these experiments.

(d) A method labelled fmax=2 uses the first subdocument of max=2 items. This strategy will use quality items (b) but also include the beginning portion of documents (c) about twice as long, and would remedy the fact that there may not be sufficient quality items for training.

3.2 Subdocument Selection Based on a Ranking Operation

These methods do a subdocument ranking operation with the routing queries first so that we can select the best ranking units for training. By design, best ranking subdocuments have high probability of being 'truely relevant' to their queries and have been proven to work in user relevance feedback. By ignoring poorer ranked units one hopes to suppress the noise portions of documents for training. A drawback in this case is that the best ranked subdocuments by default share many or high-weighted terms with a query, so that learning may become limited to enhancing the given free-text representation of the query. Subdocuments that are relevant but do not resemble the query (and therefore are not ranked early) will not be used. Performing a ranking is also time-consuming compared with methods in Section 3.1. We have experimented with two methods as given below:

(e) Select the bestn best-ranked relevant subdocuments for training after ranking with respect to the given routing query representations. A variant of this method is to enhance/expand the query representations first by using method (b) max=1 documents before doing the ranking. Selecting these bestnx best-ranked subdocuments would include more 'truely relevant' ones than before because the ranking operation is more sophisticated and has been shown to achieve improved performance in our initial TREC2 experiments [8].

(f) Select the topn highest ranked subdocuments of every relevant. Since our purpose is try to avoid noise portions of relevant documents, these top ranked units should have high probability that they are mostly the signal portions as in (e). Moreover, because all relevant documents are used, this method may include the advantage of Section 3.1 that units not resembling the query would also be included for training. A variant is, as before, to enhance/expand the queries first before ranking for the topnx highest ranked subdocuments for later training.

3.3 Subdocument Selection Based on Combination of Methods

By combining training document sets obtained from the best of the previous two subsections, we hope to improve on the individual approaches alone. Our objective is to define a training set of subdocuments that are specific to and resemble a query representation, as well as including overall subdocuments that are relevant. The following two methods have been tried:

(g) Merge documents obtained by method (e) bestn/bestnx retrieved, with those of method (b) using max=1. The rationale is that method (e) selects the best of those resembling the query, and method (b) uses short quality relevant documents in general.

(h) Merge documents obtained by method (e) bestn/bestnx retrieved, with those of method (f) topn/topnx=1 units of every document. This is similar to (g), except that instead of using short documents only, we now incorporate the best portions of every relevant.

4. EXPERIMENTS AND DISCUSSION OF RESULTS

For testing our various strategies of subdocument selection for training, we performed experiments exactly as those of TREC2 routing: Topics 51-100 retrieving on the 1 GB of documents on Disk3 of the TREC collection. Topics 51-100 have relevant document information from Disk 1&2 totaling 2 GB. There are altogether 16400 relevant documents averaging out to 328 per query. During our processing however, a small percentage of the relevants are lost, so that we in effect use only 16114 relevants that get segmented into 57751 subdocuments. This averages to about 1155 units per query. For the ranking strategies of Section 3.2, we have created a separate subcollection consisting only of the 57751 training relevants but using Disk 1&2 term statistics, and ranking for the first 2000 of each query is done. Various subsets of these ranked training documents are then used for weight learning for the query-term side of the network, with term expansion level K=40 terms as the standard. For some cases we also did term expansion of K=80. After freezing these trained edge weights, Disk3 subdocuments are linked in and routing retrievals are done. Results using the 'total number of relevants retrieved' (at 1000 retrieved cutoff) and 'average precision over all recall points' as measures of effectiveness, as well as the number of training units used, are summarized in Table 1. Some of the detailed precision-recall values are given in Table 2. The overall conclusion from these results is that for this TREC-2 routing experiment, where a large number of relevant documents of different sizes and quality is available, it is possible to define good subsets of the documents or portions of them for training.

From Table 1 and using the average precision (av-p) measure for comparison, it appears that the simple strategy (b) of just using short, 'nonbreak' max=1 relevant documents gives one of the best results, achieving av-p at K=40 expansion level of 0.4050, about 6.7% better than the 0.3795 of our baseline strategy (a) which uses all the relevant units. Moreover it is very efficient, requiring only 5235 units which is less than 10% of the total 57751 relevant subdocuments available and about 1/3 of the 16114 documents. Using longer documents that break into two and six units (max=2 and 6) successively leads to slightly worse results as well as more work (15103 and 32312 subdocuments). Thus, it appears that longer documents carry with it more noise as discussed in the Introduction. Just using the first subdocument of every relevant (c) performs quite well, with av-p of 0.4001. Since the FR collection has many documents of thousands of words long, it is difficult to imagine that signal parts are all in the first subdocuments. A casual scan however shows that some FR documents, such as FR88107-0009 and FR88119-0018, carry a summary at the beginning. Moreover, FR documents constitute only a minority of the training relevants. Thus the first subdocuments apparently carry sufficient signals of documents for training in this experiment. Last subdocuments (results not shown) do not perform as well as first. One of the best results is fmax=2 achieving av-p of 0.4047 as good as 'nonbreak' max=1 method and using 10,169 training units.

Surprisingly, using the best ranking bestnx=30, 100, 300, 2000 subdocuments (e) gives 0.3790, 0.3993, 0.3999 and 0.3877 average precision respectively, peaking around bestnx=300 but does not give better performance than (b,c,d) strategies. For bestnx=30, employing only 1500 subdocuments apparently is not sufficient, and training may be limited to subdocuments resembling the original query. bestnx=100 uses 4945 units similar to max=1 but with av-p about 1.5% worse, while bestnx=300 uses 13712 which is slightly less than first and performs about the same. In general, bestn results (not shown) are slightly less than those of bestnx as expected. Using the topnx=1 subdocument of every relevant (f) achieves 0.4082, the best numerically. In (f) we have less than 16114 units for training because we only rank the top 2000 for each query, and so some subdocuments ranking below 2000 are not accounted for. It appears that including other overall relevants can help improve performance.

Strategies (g,h) of combining sets of subdocuments do not seem to lead to more improved results.

Using the relevants retrieved (r-r) as a measure, it appears that larger training set sizes between 10000 to 16000 are needed to achieve good recall. For example, max=1 and bestnx=100 employs about 5000 units for training and have r-r of 7646 and 7605. bestnx=300, max=2, first and topnx=1 have r-r values of 7703, 7783, 7805 and 7833, and training set sizes of: 13712, 15103, 16114 and 15702. fmax=2 achieves good r-r of 7827 with a training size of 10169. fmax=3 (results not shown) is inferior. For this collection, the best strategies of selecting subdocuments for training appears to be either fmax=2 with av-p/r-r values of 0.4047/7827 or topnx=1 with 0.4082/7833. fmax=2 has the advantage that a ranking is not done and the training set is smaller. The detailed recall-precision values in Table 3 also shows that fmax=2 gives better precision at the low recall region. It appears that using document properties to select training documents in this routing experiment is both effective and efficient.

5. CONCLUSION

We explore several strategies of selecting relevant documents or portions of them for query training in the TREC-2 routing retrieval experiment. It confirms that using all relevants for training is not a good strategy because irrelevant noisy portions of documents would be included. Short relevants are the quality documents. Simple methods such as using only short documents, together with beginning portions of longer documents for training performs well and is also efficient. For this TREC2 routing, an average of about 200-300 subdocuments per query appears adequate, about 1/5-1/4 of all known relevant subdocuments available in this experiment. Selecting the bestn ranked relevants (as in relevance feedback) is not as effective as just selecting the top ranked unit of every document. This investigation also shows that breaking documents into subdocuments is useful for query training.

ACKNOWLEDGMENT

This work is partially supported by a grant from ARPA via the TREC program.

REFERENCES

1. Salton, G. & Buckley, C. Improving retrieval performance by relevance feedback. J. of American Society for Information Science. 41 (1990), 288-297.

2. Harman, D. Relevance feedback revisited. In: Proc. ACM SIGIR 15th Ann. Intl.Conf. on R&D in IR. Belkin, N.J, Ingwersen, P & Pejtersen, A.M (Eds.) ACM, NY. (1992), 1-10.

3. Kwok, K.L., Papadopolous, L. & Kwan, Y.Y. Retrieval experiments with a large collection using PIRCS. In: NIST Special Publication 500-267. Gaithersburg, M.D. 20899. (March 1993), 153-172.

4. Haines, D & Croft, W.B. Relevance feedback and inference networks. In: Proc. ACM SIGIR 16th Ann. Intl.Conf. on R&D in IR. Korfhage, R, Rasmussen, E & Willet, P (Eds.) ACM, NY. (1993), 1-11.

5. Harman, D (Ed.) The First Text REtrieval Conference (TREC-1). National Institute of Standards and Technology Special Publication 500-207, March 1993.

6. Kwok, K.L. A network approach to probabilistic information retrieval. Submitted for publication.

7. Harman, D (Ed.) The Second Text REtrieval Conference (TREC-2). National Institute of Standards and Technology Special Publication, to be published.

8. Kwok, K.L., Grunfeld, L. TREC-2 retrieval experiments using PIRCS. In: NIST Special Publication, to be published.

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

10. van Rijsbergen, C.J. Information Retrieval, Second Edition. Butterworths, London. (1979).

11. Kwok, K.L. Experiments with a component theory of probabilistic information retrieval based on single terms as document components. ACM Trans. on Information Systems. 8 (1990), 363-386.

12. Kwok, K.L. A neural network for probabilistic information retrieval. In: Proc. ACM SIGIR 12th Ann. Intl. Conf. on R&D in IR. Belkin, N.J. & van Rijsbergen, C.J. (Eds.) ACM, NY. (1989), 21-30.

	------------- Expansion Level K -----------				
	40		80	No. of Training	
	r-r/av-p	%inc	r-r/av-p	Subdocs	%
a) all relev subdocs	7611/.3795	baseline	7563/.3746	57751	baseline
b) max=1	7646/.4050	0.5/6.7	7695/.4084	5235	9
=2	7783/.3970	2.3/4.6		15103	26
=6	7762/.3891	2.0/2.5		32312	56
c) first	7805/.4001	2.5/5.4	7854/.3976	16114	28
d) fmax=2	**7827/.4047**	**2.8/6.6**	**7861/.4040**	**10169**	**18**
e) Best Ranked					
bestnx=30	7295/.3790	-4.2/-0.1		1500	3
=100	7605/.3993	-0.1/5.2		4945	9
=300	7703/.3999	1.2/5.4		13809	24
=2000	7739/.3877	1.7/2.2		31792	55
f) Top subdoc					
topn =1	7821/.4067	2.8/7.2		15384	27
topnx=1	**7833/.4082**	**2.9/7.6**	**7887/.4062**	**15702**	**27**
g) Merge Max=1,bestn					
mbestn100	7743/.4053	1.7/6.8		8930	15
h) Merge topn=1,bestn					
tbestn100	7798/.4069	2.5/7.2		16362	28

Table 1: Relevants Retrieved (r-r), Average Precision Values (av-p) and Number of Training Subdocuments for Various Subdocument Selection Strategies

Strategy:	**all**	**max=1**	**first**	**fmax=2**	**bestnx=300**	**topnx=1**
Interpolated Recall - Precision Averages:						
0.0	.8311	.8475	.8362	.8467	.8273	.8404
0.1	.6464	.6751	.6779	.6839	.6664	.6808
0.2	.5755	.6116	.5978	.6132	.6000	.6086
0.3	.5035	.5413	.5285	.5312	.5240	.5429
0.4	.4469	.4774	.4734	.4786	.4719	.4810
0.5	.3951	.4288	.4245	.4245	.4206	.4259
0.6	.3286	.3681	.3564	.3565	.3641	.3633
0.7	.2706	.2880	.2833	.2880	.2830	.2904
0.8	.2057	.1937	.2085	.2099	.2095	.2182
0.9	.1079	.1144	.1156	.1181	.1159	.1183
1.0	.0115	.0107	.0120	.0135	.0113	.0123
Average precision (non-interpolated) over all rel docs						
	.3795	**.4050**	**.4001**	**.4047**	**.3999**	**.4082**
Precision:						
At 5 docs	.6480	.7160	.6920	.7120	.6920	.6920
10 "	.6460	.6860	.6940	.6968	.6820	.6960
20 "	.6100	.6540	.6540	.6670	.6520	.6520
100 "	**.4706**	**.4930**	**.4854**	**.4890**	**.4970**	**.4926**
500 "	.2439	.2490	.2532	.2524	.2493	.2544
1000 "	.1522	.1529	.1561	.1565	.1541	.1567
R-Precision (precision after R (=num_rel for a query) docs retrieved):						
Exact	**.4036**	**.4283**	**.4218**	**.4228**	**.4201**	**.4274**

Table 2: Average Precision Values at Interpolated Recall Points and at Number of Documents Retrieved for Six Subdocument Selection Strategies (Expansion Level=40)