THUIR at TREC 2003: Novelty, Robust and Web*

Min Zhang, Chuan Lin, Yiqun Liu, Le Zhao, Shaoping Ma
State Key Lab of Intelligent Tech. & Sys., CST Dept, Tsinghua University, Beijing 100084, China
z-m@tsinghua.edu.cn

This is the second time that Tsinghua University Information Retrieval Group (THUIR) participates in TREC. In this year, we took part in four tracks: novelty, robust, web and HARD, describing in following sections, respectively. A new IR system named TMiner has been built on which all experiments have been performed. In the system, Primary Feature Model (PFM)[1] has been proposed and combined with BM2500 term weighting [2], which led to encouraging results. Word-pair searching has also been performed and improves system precision. Both approaches are described in robust experiments (section 2), and they were also used in web track experiments.

1. Novelty track

Our research on this year's novelty track mainly focused on four aspects: (1) unsupervised relevance judgment; (2) efficient sentence redundancy computing; (3) supervised sentence classification; (4) supervised redundancy threshold learning.

1.1. Unsupervised relevance judgment

The work of finding relevant information is useful for task1 and task3. Since words mismatch problem is dominant in sentence comparison, three kinds of approaches have been carried out in unsupervised relevance judgment to solve the problem as following.

(1) Query Expansion (QE) using WordNet synonymy and hyponomy, and Dr Lin Dekang's dictionary of term dependency [3];

(2) QE with query terms' local co-occurrence words in a window of W according to supposed relevant document set (without initial search, named LCE in our previous study[4]). In task3, the co-occurrence information was got in given relevant sentences in top five documents;

(3) Pseudo relevance feedback. After initial retrieval, top M terms in top ranked N_1 sentences and not in last N_2 sentences were added to the query. In task3, the top documents were defined as given relevant sentences.

Approach (1) and (2) had been already described in our last year's novelty track report [4], and were to make further observation in this year's tasks. Therefore we only give some more information about approach (3) as follows.

Local feedback strategies are based on expanding the query with terms correlated to known query terms. Generally, the expanding n terms are extracted from the top m document (In our novelty track experiments, each sentence is taken as an individual document, m=2) after initial search. The n terms are chosen based on the similarity sim(q, k_v) of them[5]. The value of sim(q, k_v) is calculated as:

$$sim(q,k_v) = \vec{q} \cdot \vec{k_v} = \sum_{k_u \in Q} w_{u,q} \times c_{u,v}$$

Where $w_{u,q}$ is indexed query weight. $c_{u,v}$ is calculated as follows in our experiments:

* Supported by the Chinese National Key Foundation Research & Development Plan (Grant G1998030509), Natural Science Foundation No.60223004, and National 863 High Technology Project No. 2001AA114082.

i. $c_{u,v} = \sum_{d_j \in D_l} f_{s_u,j} \times f_{s_v,j}$,

where $f_{s_u,j}$ and $f_{s_v,j}$ are frequencies that term s_u and s_v occurred in the whole document, respectively.

ii. $c_{u,v} = \sum_{d_j \in D_l} f'_{s_u,j} \times f'_{s_v,j}$

where $f'_{s_u,j}$ and $f'_{s_v,j}$ are frequencies that term s_u and s_v occurred within the distance of W words to the observing query term, respectively. In our experiments, the window W is set to 3.

iii. $c_{u,v} = \sum_{k_i \in V(s_u)} \sum_{k_j \in V(s_v)} \frac{1}{r(k_i, k_j)}$

Where $r(k_i, k_j)$ is the distance between terms k_i and k_j in the same sentence. If k_i and k_j are in different documents, $r(k_i, k_j) = \infty$.

There are quite a few terms that occur in a great deal of sentences frequently. They may be in both relevant and irrelevant sentences. We assume that the last k sentences in the initial retrieval are not relevant (k=25 in all experiments), therefore terms in these sentences are useless and should not be expanded.

Table 1 shows experimental results with above approaches. All QE were performed on short queries. It is shown that all QE approaches improve the system performance. The more terms expanded, the better results were got. Among all QE approaches we observed, local co-occurrence within 10 words' window with Mutual Information weighting performs best, which made 14.5% improvement. And QE with Dr Lin Dekang's proximity and dependency dictionary are also greatly helpful. Unfortunately, our official run THUIRnv0312 used the one with almost the least improvement.

Table1 experimental results of QE with short query in task 1

	QE approaches	P	R	F	Improvement (vs short)
No QE	Short query	0.633	0.637	0.552	--
	Long query	**0.593**	**0.805**	**0.622**	**+12.7%**
WordNet	synset	0.625	0.665	0.564	+2.17%
	Hyponomy	0.618	0.68	0.569	+3.08%
	Synset + hyponomy	0.616	0.695	0.575	+4.17%
Dr Lin dekang's Dictionary	**proximity**	**0.58**	**0.831**	**0.608**	**+10.1%**
	dependency	**0.59**	**0.827**	**0.616**	**+11.6%**
Local Cooccurrence Expansion	MI,win = 1	0.557	0.866	0.613	+11.1%
	MI,win=1,sim>0.1	0.632	0.638	0.552	0%
	MI,win = 10	**0.536**	**0.955**	**0.632**	**+14.5%**
Pseudo Feedback	Top 10 terms	0.593	0.716	0.584	+5.8%
	Top 15 terms	0.59	0.732	0.587	+6.34%
	Top 100 terms	0.589	0.744	0.594	+7.61%
Combine (Official run)	***Syn+mi_win1_sim>0.1***	***0.624***	***0.665***	***0.564***	***+2.17%***

1.2. Efficient sentence redundancy computing

On sentence redundancy elimination, rather than general similarity computing, we used

unsymmetrical sentence overlap. That is the same as last year. Our experimental results show it makes trivial improvement comparing to the symmetrical measure of similarity.

Besides, the subtopic-based redundancy elimination has been proposed. Generally in a topic, several key-points are concerned, while only one point can be described in each result sentence. Therefore, a natural idea is to divide the topic into subtopics, and then inter-topic documents, inborn, take novel information. Therefore only documents in the same subtopic (cluster) should be used to calculate redundancy. Considering that clusters will help prevent the elimination of inter-cluster documents, if clusters are neatly defined, a better recall will be assured.

Three subtopic clustering approaches have been proposed: one is topic-oriented, and another two are document-oriented.

For topic-oriented clustering approach, <title>, <description>, and each sentence in "narrative" were taken as subtopic individually. Documents were clustered to subtopics according to their distance to each subtopic description. (Shown as subtopic I in following)

For document-oriented clustering, subtopic clusters are built by two ways:

(1) Clustering with syntax analysis: To "event" type topic, take date (year and month) as the feature; to "opinion" type topic, the opinion holder is taken as the feature. Both features were extracted using rules automatically. (Shown as subtopic II in following)

(2) Automatic results clustering by sentence vector distance using KNN-like approach. The sentences vectors are extremely sparse so that only a small number of clusters were formed. Therefore, the method hardly improves (though not deteriorating) the results. (Shown as subtopic III in following)

Table 2 shows the effects of different subtopic-based redundancy elimination approaches. All official runs we submitted in task1 were using overlap-based redundancy computing, and the elimination thresholds were all set to 0.55 in task1.

In task2, mistakes has been made during upload the official runs results because of the coming of submit time deadline. Then all official runs are all wrong in task2, which we can not resume and the THUIRnv0221 and THUIRnv0222 are the same ones. The correct ones we tend to submit are also listed in Table2 (un-official runs).

Table 2 Results of subtopic-based redundancy elimination (task2)

Task		P	R	F	
1	Baseline, short query, overlap threshold = 0.55	0.49	0.58	0.453	THUIRnv0313
	Subtopic I, threshold = 0.55	**0.46**	**0.74**	**0.505**	**THUIRnv0315**
	Short query, Subtopic II, threshold = 0.8	0.47	0.62	0.457	un-official run
	Short query, Subtopic III, threshold = 0.8	0.48	0.62	0.458	un-official run
2	Subtopic II, threshold = 0.8	0.708	0.983	0.812	un-official run
	Subtopic III, 4 clusters, threshold = 0.8	**0.713**	**0.982**	**0.815**	**un-official run**
	Baseline, tuned threshold = 0.8	**0.714**	**0.980**	**0.815**	**un-official run**
	Unknown wrong submission	0.68	0.72	0.687	THUIRnv0323
	Unknown wrong submission	0.69	0.69	0.679	THUIRnv0321 & THUIRnv0322

Besides above approaches, redundancy computing based on triple overlap has been proposed and studied. Firstly, extract syntax triples of each relevant sentence. Secondly, only those triples of V or N are kept for further computation. Thirdly, compute overlap of each sentence pair by triples' overlap. At last, give a threshold and eliminate sentences with high overlap score. Following table3 shows the

effects of the approach. The approach was used in task4, and the threshold for official runs is learnt by the given five documents. Following Table3 shows the effects of the approach.

Table 3 Results of triple-overlap-based elimination

Elimination threshold	P	R	P*R	F	
0.8	0.716	0.789	0.563	0.734	un-official run
0.85	*0.70*	*0.88*	*0.614*	*0.765*	*THUIRnv0342*
0.9	**0.684**	**0.939**	**0.641**	**0.777**	**un-official run**
0.95	0.652	0.982	0.641	0.771	un-official run
Different threshold for Each topic	0.720	0.780	-	0.728	*THUIRnv0341*
Different threshold for Each topic (tuned parameters)	**0.728**	**0.954**	**0.697**	**0.819**	**un-official run**

1.3. Supervised sentence classification

In task3, we taken the problem of finding relevant information as finding a suitable binary sentence classification using provided relevant sentences in top five documents. A SVM classifier has been used. Features were extracted according to key words in training sentences (sentences given in first five documents). The basic assumption is that the features of relevant sentences are different from that of irrelevant sentences. In terms of positive examples and negative examples, SVM finds a hyper-plane in the feature space, which is chosen to maximize the margin of the training positive and negative points [6]. In the given first 5 documents of each topic, relevant sentences are used as positive examples, and the remaining ones are negative examples. Proportion of positive and negative examples has been balanced in our approach by giving a weighting of 150:1. A linear SVM has been trained. We used a SVM package (version 2.4, by Chih-Wei Hsu, etc. [7]) to create the classifier.

It shows that this supervised sentence classification does helpful on finding relevance, which can be seen in the following Table 4.

Table 4 Effects of supervised sentence classification using SVM

	P	R	F	
Baseline (using long query)	0.43	0.73	0.479	THUIRnv0334
Linear SVM	**0.42**	**0.82**	**0.493**	**THUIRnv0331**

1.4. Supervised redundancy threshold learning

As we known, redundancy threshold setting is one of the important issues in finding new information. For unsupervised task (task1 and task2), according to the high similarity between documents and topics, a fixed threshold has been set. While For task3 and task4, supervised learning has been performed. We trained the threshold by two ways: one is to tune a fixed threshold to all topics by the given known documents, and another one is to tune a different parameter for each topic. It seems that the elimination threshold trained by top five documents, which was set to fixed 0.8 in task4, works well to the remaining ones.documents. Further analysis and observation need to be done.

1.5. Submitted official runs

Task	Approach description	P	R	F	
1rel	Short query+feedback	0.61	0.73	0.593	THUIRnv0311
	Synset + LCE(MI, win = 1, sim>0.1)	0.62	0.67	0.564	THUIRnv0312
	Baseline, short query	0.49	0.58	0.453	THUIRnv0313
	cut results when sim<0.95*top_sim	0.63	0.63	0.548	THUIRnv0314
	Subtopic I	0.46	0.74	0.505	THUIRnv0315
1new	Short query+feedback, overlap, threshold=0.55	0.47	0.66	0.481	THUIRnv0311
	Synset + LCE(MI, win = 1, sim>0.1)	0.48	0.61	0.459	THUIRnv0312
	Baseline, short query, overlap threshold = 0.55	0.49	0.58	0.453	THUIRnv0313
	cut results when sim<0.95*top_sim	0.49	0.57	0.451	THUIRnv0314
	Subtopic I, overlap threshold = 0.55	0.46	0.74	0.505	THUIRnv0315
2new	Unknown wrong submission	0.69	0.69	0.679	THUIRnv0321 & THUIRnv0322
	Unknown wrong submission	0.68	0.72	0.687	THUIRnv0323
3rel	SVM classification	0.57	0.87	0.627	THUIRnv0331
	Long query + feedback	0.56	0.89	0.623	THUIRnv0332
	Long query + QE with LCE (window=10)	0.54	0.92	0.621	THUIRnv0333
	Long query	0.58	0.81	0.610	THUIRnv0334
3new	SVM classification, overlap, threshold=0.8	0.42	0.82	0.493	THUIRnv0331
	Long query + feedback，threshold=0.85	0.40	0.86	0.489	THUIRnv0332
	Long query + LCE (win=10), threshold=0.8	0.39	0.88	0.491	THUIRnv0333
	Long query, topic-different threshold	0.43	0.73	0.479	THUIRnv0335
4new	Triple overlap with topic different threshold	0.72	0.78	0.728	THUIRnv0341
	Triple overlap with fixed threshold 0.85	0.70	0.88	0.765	THUIRnv0342
	Overlap, threshold = 0.8	0.68	0.99	0.791	THUIRnv0343
	Overlap + QE (MI, win=1,sim>0.8)	0.68	0.98	0.790	THUIRnv0344
	Overlap + event/opinion clustering, threshold0.8	0.68	0.96	0.780	THUIRnv0345

2. Robust track

In this year's robust track, our basic idea is: "bad" topics are not topics that only seldom documents can be returned after retrieval, but the ones that return too many irrelevant documents. Therefore the key point of our work is to improve the system precision to these topics, namely, to perform a cagey and strict judgment of relevance.

We implemented two novel approaches in our TMiner system, namely Primary Feature Model and Word Pair Search, aiming to enhance system performance in terms of precision. They are used in both robust and web tracks.

Besides, query quality is also an important factor to system performance. We eliminate the negative description of the topic in <narr> field automatically, hence a long query without negative information was generated. Effect of using other fields of topics was also observed, shown in Figure1.

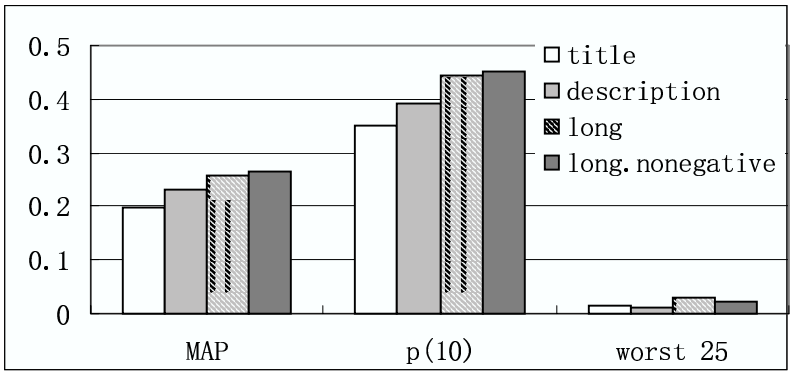

Figure 1 Effects of using different topic field to generate queries

2.1. Primary Feature Model

As we known, terms appear in the title, heading or other emphasized fields in the document are more generally important to the reader than the other body text. They represented the notion of the authors on the main content of the document. We take these special fields as Primary Feature Fields (*PFF*). Terms in the primary feature fields in the entire collection construct a Primary Feature Space (*PFS*). This space is not of uniform distribution. Therefore, terms with higher density in the space should be more significant as features. Generally, the density of the feature dimension can be represented by the term frequency in the primary feature space:

$$D_i = \sum_{k=1}^{N} tf_{ik}$$

Where tf_{ik} is occurrence frequency of term i in *PFF* of document k, and N is the total number of documents in the collection.

Since documents are built by different authors with different backgrounds and writing habits, the description of the field may not be canonical. To reduce the influence of the information leak caused by different author, it's better to limit the same terms from one page contributing to the feature density only once. i.e.

$$tf_{ik} = \begin{cases} 1, & if\ term\ i\ occurs\ \ in\ the\ primary\ field\ of\ \ doc\ k \\ 0, & otherwise \end{cases}$$

Therefore the density of the feature dimension is be simplified to:

$$D_i = \sum_{k=1}^{N} tf_{ik} = n_i$$

Where n_i is the number of documents which contains term i in its primary feature space in the collection. Then the term weight in the primary feature space can be represented by:

$$w^{(2)} = \log(n+1)$$

Where tf_i and n_i are simplified as tf and n respectively.

By doing so, in a documents collection, the weight of a term in the query is composed of the weights in *PFS* and in body text. Therefore the new similarity scoring function can be presented as:

$$\sum_{T \in Q} [\lambda w_{BT}^{(1)} + (1-\lambda) w_{PF}^{(2)}] \frac{(k_1+1)tf}{(K+tf)} \frac{(k_3+1)qtf}{(k_3+qtf)}$$

where $w_{BT}^{(1)}$ is the Robertson/Sparck Jones weight for a term according to body text, $w_{PF}^{(2)}$ is the weight in terms of primary feature field, λ is the impact factor of the body text. When λ =1, the scoring function is same as traditional Robertson/Sparck Jones probabilistic model.

In this year's robust task, the <headline> of the documents were used as Primary Feature Field. The effects are shown in Table 5 (λ=0.1). Trivial improvement is got on the performance of worst topics.

2.2. Word Pair Search

The basic idea of word pair is that if adjacent words in the query are also adjacent in the document, then the document would be more likely to be relevant. In our word pair implementation, word pairs in a query are treated as additional terms for the query; and *DF*, *TF* information are calculated and added to the original query term weights by a weighted summation.

$$W = \sum_{t \in query\ terms} W^{(1)} \frac{(k_1+1)\, tf_t\, (k_3+1)\, qtf_t}{(K+tf_t)\,(k_3+qtf_t)} + \lambda \sum_{wp \in query\ wordpairs} W_2$$

$$where \quad W^{(1)} = \log \frac{N - df_t + 0.5}{df_t + 0.5}, \qquad W_2 = W_2^{(1)} \frac{(k_1+1)\, tf_{wp}\, (k_3+1)\, qtf_{wp}}{(K+tf_{wp})\,(k_3+qtf_{wp})}, \qquad W_2^{(1)} = \log \frac{N - df_{wp} + 0.5}{df_{wp} + 0.5}$$

Note: a weight λ is multiplied to the word pair part of the total weights to constrain the impact of the word pair weight on the final document results.

DF of word pair (e.g. word pair *AB*) can be estimated in different ways:

a. *DF* is specified by a constant the user provides. This estimation method is called wp_1 in our experiments.
b. *DF* of *AB* can be estimated with the number of documents that contain both the two terms: *A* and *B*. We call it wp_2 method.
c. *DF* of *AB* can be the exact document frequency of the pair *AB*. The wp_3.

Effects of Word Pair Search are shown in Table 6. It is encouraging that although Word Pair Search makes much improvement in terms of average system performance, but it enhances the average performances of "bad" topics obviously. (In the table 6, λ of title and description query were set to 0.2 and 0.3 respectively).

Table 5 Effects on using Primary Feature Model

	MAP	p(10)	worst 25 topics
long	0.2571	0.444	0.0282
long.pfs	0.2597	0.446	0.0289
long.non-neg	0.2667	0.452	0.0235
long.no-neg.pfs	0.2676	0.453	0.0252
description	0.2307	0.392	0.0114
description.pfs	0.2339	0.394	0.0118

Table 6 Effects of Word Pair Search

	MAP	p(10)	worst 25 topics
Title	0.199	0.351	0.0142
Title.wp_3	0.205	0.356	0.0153
	+3.1%	**+1.4%**	**+7.8%**
Desc	0.231	0.392	0.0114
Desc.wp_3	0.240	0.406	0.0132
	+3.8%	**+3.6%**	**+15.8%**

2.3. Submitted Official Runs

		MAP (%)			p(10) (%)			worst 1/4 topics (%)		
		old	new	all	old	new	all	old	new	all
1	**THUIRr0301**	**13.73**	**38.21**	**25.97**	**36.00**	**53.20**	**44.60**	**2.03**	**5.70**	**2.89**
2	THUIRr0302	14.45	38.88	26.67	36.20	54.20	45.20	1.61	5.65	2.35
3	THUIRr0303	13.31	38.00	25.71	35.00	53.00	44.00	1.99	5.61	2.84
4	THUIRr0304	12.37	37.06	24.72	31.20	50.40	40.80	1.52	4.41	1.95
5	THUIRr0305	11.66	37.02	24.34	31.20	50.80	41.00	0.94	4.75	1.48

1: long query retrieval, using BM2500+PFS weighting

2: long query without negative information, using BM2500+PFS weighting

3: long query baseline retrieval

4: combine short query retrieval and retrieval on description field

5: retrieval on description field with Word Pair Search and BM2500 + PFS weighting

3. Web track

3.1. Introduction

In this year's TREC Web Track research, we participated in both Known-Item Search and Topic Distillation Tasks.

In Topic Distillation task, only entry pages of key sites should be returned as results. Therefore, two kinds of approaches have been studied: "Top-to-Bottom" retrieval, and "Bottom-to-Top" one. The key point in both approaches is: how to give a clear and precise definition of entry pages and use it to get an entry page list. According to our definition, we developed an entry page locating algorithm concerning the following characteristics of pages: URL information, document length, in-site out-link rate and in-link number. With the algorithm, four entry page lists have been built according to different threshold (strict or relaxed metric) and number of entry pages returned from one site.

On term weighting, a Primary Feature Space (PFS) model has been proposed. In PFS model, as has been described in section 2, a DF-related weighting is performed on words in Primary Feature Field (PFF), which makes improvement compared with traditional IDF-related weighting. In-link anchor text, title and keywords of in-site-out-link pages, and bold text are proved to be good PFF in our experiments.

Besides, searching with word pair has been added to scoring, which improves the accuracy of result ranking, as has been described in section 2.

A novel site uniting method is also proposed to confirm that any key resource we found is not part of a larger site also principally devoted to the topic.

In Page Finding task, other than in-link anchor text used in TD task, the out-link anchor text are used to decide the webpage candidate set. Then rank the candidates according to the term weighting on full text and return top results. It shows that the out-link anchor text is much more effective than the full text of the page. And it is different to the general idea that the out-link anchor text describes the page the link point to more than the page itself.

Abbreviations were extracted from the corpus automatically, and were expanded in topics.

3.2. Definition and locating algorithm of entry pages

In this year's topic distillation task, we are concentrating solely on websites (represented by their entry pages) as key resources. Before retrieval, it is necessary for us to find out what an entry page is

and how to find it. In our dictionary, an entry page is the main entry point of a web site and a bridge between pages inside and outside the web site; it should have the following characteristics:

(a) An entry page can connect (directly or indirectly) to all pages in the web site it belongs to.

(b) When pages outside the web site want to visit pages inside the site, they mainly visit the entry page at first.

According to the definition, we may see that entry pages should have some special features in their URLs, link structures, etc. In our research, we divide entry pages into two types: topic-related and non-topic-related. For topic-related entry pages, their URLs always contain one or more query words. For example, “www.drugabuse.gov/DrugPages/Marijuana.html” is an entry page of its sub-site for the topic “Where can I locate information on the pros and cons of legalizing marijuana”. For non-topic-related entry pages, they always have the following features:

(1) url: root, subroot, named as "index.htm" etc, or named as the topic words of the sub-site;

(2) length: an entry page is usually not too long;

(3) in-site-out-link: an entry page should have enough in-site-out-link;

(4) in-link: an entry page should have enough in-link (especially links from pages in different site/server).

In this way, about 15% pages have been selected to form the key entry pages collection, which cover about 68% useful key resources for 50 testing queries.

3.3. Topic distillation with entry page lists

With the entry page locating algorithm, four entry page lists have been built according to different threshold (strict or relaxed metric) and number of entry pages returned from one site.

List Name	Description
Base list	Strict threshold
Unite list	Loose threshold, but only one Entry Page per site
Parse list	Strict threshold, and only one Entry Page per site
Root list	Non-file pages only, and one Entry Page per site

1) Top-down topic distillation

In the top-down algorithm, we restrict our data set on selected entry pages. With an ideal entry page locating algorithm, entry page retrieval is obviously a clever choice for topic distillation task, because it discard pages that are not possibly key resources. But in practical world, any locating algorithm brings in noises and may discard some entry pages which will perhaps be key site representatives.

2) **Bottom-up topic distillation**

In the bottom-up algorithm, we try to locate key sites instead of key pages. Entry page of one site is returned as result. We first retrieve on full text, and then use the following algorithm to re-rank the results:

```
For each item in the Entry Page list {
        For each page N an Entry Page A links to {
            weight (A) += weight (N)
        }
}
Re-rank the result list;
```

3) Comparison with ordinary full text-retrieval

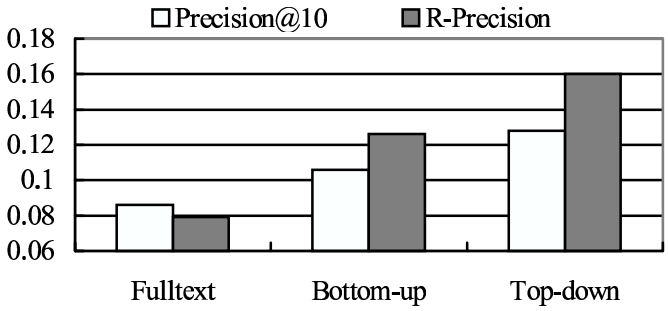

Figure 2 bottom-up and top-down comparison

According to experiment results in Figure 2, both bottom-up and top-down perform much better than full-text retrieval. It shows the entry page locating algorithm is conceive and reliable. And the top-down method is perhaps more useful for this kind of topic distillation.

3.4. Using document structures

The effect of using HTML document structure is studied in our experiments. It is found that in-link anchor information is useful for known-item search. We also tested several new parts of HTML document in topic distillation experiments. They are: in-site out-link anchors, in-site out-link page titles and in-site out-link page keywords.

For a certain page A, in-site out-link anchors are anchors describing links from A to other pages in the site where A is. So the anchors are located on A; that is quite different from frequently-used in-link anchors which are on pages linking to A. However, in-site out-link title / keyword are on pages A points to, although in the entry page finding process, we can consider them as descriptions of A.

Using of in-site out-link is particularly useful for topic distillation task, shown in Figure 3, the experiment is on the entry page data set. Full text retrieval experiment is shown in Figure 4.

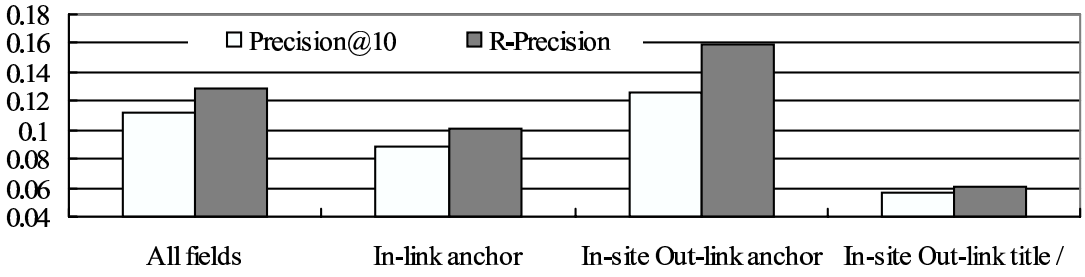

Figure 3 Fields retrieval on entry page text set

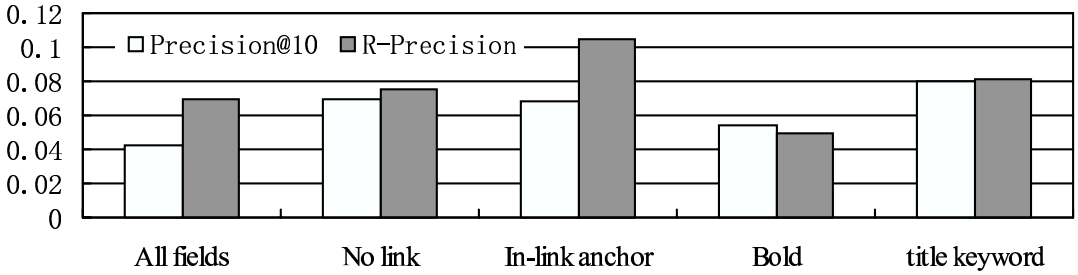

Figure 4 Fields retrieval on full text set

Then we can conclude that:

To the whole collection, in-link anchor performs better. To the entry pages text set: 1) in-link anchor no longer works; 2) full text is reliable; 3) in-site out-link anchor leads to astonishing results; 4) in-site title / keywords may be too short to improve retrieval results.

3.5. Topic distillation with different weighting schemes

We experimented with the following weighting schemes in the topic distillation task: Primary feature space model and BM2500 model. In the chart below, category axis represents the rate BM2500/PFS. It shows that with a broad parameter scale, we get improvement using PFS weighting scheme combined with BM2500 weighting.

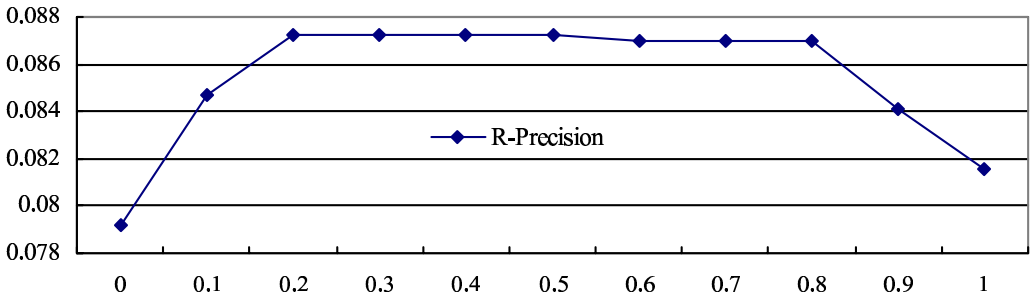

Figure 5 PFS + BM2500 retrieval

We compared the two weighting schemes in the following charts. The first experiment is performed on full text data set; while the second is on the entry page set. It shows that PFS method only performs poor, but it can stably improve performance if we combines it with the BM2500 weighting scheme.

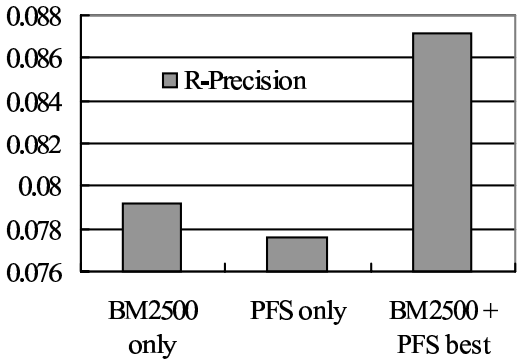

Figure 6 weight schmes on full text set

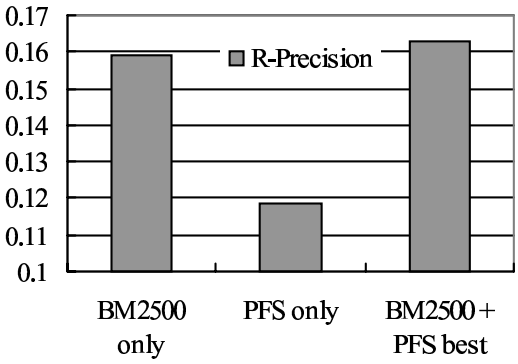

Figure 7 weight schmes on entry page set

3.6. Runs submitted and Evaluation Results

Topic distillation runs description:

Runs	Data Set	Weighting	Field Selection	R-pre
THUIRtd0301	Parse list	BM2500 only	Full text bold	0.1036
THUIRtd0302	Unite list	BM2500 only	Full text bold	0.0994
THUIRtd0303	Base list	PFS only	Entry Page in-site out-link title / keyword	0.0786

THUIRtd0304	Root list	PFS only	Entry Page in-site out-link anchor	0.0692
THUIRtd0305	Full text	BM2500+ PFS	Full text all fields	0.1262
Unofficial run1	Base list	BM2500 only	Entry page all fields	**0.1293**
Unofficial run2	Base list	BM2500+ PFS	Entry Page in-site out-link anchor	**0.1629**

Known item search runs description and results:

Runs	Description	MRR
THUIRpf0301	Full text retrieval +NPrank+ THUIRtd0305 (BM2500)	0.561
THUIRpf0302	Full Text retrieval + THUIRtd0305 (BM2500)	0.45
THUIRpf0303	THUIRpf0304 + NPrank	0.496
THUIRpf0304	Retrieve on anchor text, using abbr., BM2500	0.463
THUIRpf0305	Retrieve on anchor text, not using abbr., BM2500	0.466

Reference

[1] Min Zhang, Ruihua Song, and Shaoping Ma, DF or IDF? On the Use of HTML Primary Feature Fields for Web IR, the 12th World Wide Web conference, (www2003), poster, May, Hungarian, 2003.

[2] Robertson, S. E., and Walker, S., Okapi/Keenbow at TREC-8. In TREC-8.

[3] http://www.cs.ualberta.ca/~lindek/downloads.htm

[4] Min Zhang et al, Expansion-Based Technologies in Finding Relevant and New Information: THU TREC2002 Novelty Track Experiments, in TREC-2002.

[5] R.Attar and A. S. Fraenkel. Local feedback in full-text retrieval systems, Journal of the ACM, 24(3): 397-417, 1977

[6] Marti A. Hearst, Trends and controversies: Support vector machine. IEEE Intelligent Systems, 13(4): 18-28, 1998.

[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html