TREC-11 Experiments at CAS-ICT: Filtering and Web

Hongbo Xu, Zhifeng Yang, Bin Wang, Bin Liu, Jun Cheng, Yue Liu, Zhe Yang, Xueqi Cheng, Shuo Bai
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{wangbin,hbxu,zfyang,yliu,cxq,yangzhe,bai}@ict.ac.cn
http://www.ict.ac.cn/

Abstract

CAS-ICT took part in the TREC conference for the second time this year and we undertook two tracks of TREC-11. For filtering track, we have submitted results of all three subtasks. In adaptive filtering, we paid more attention to undetermined documents processing, profile building and adaptation. In batch filtering and routing, a centroid-based classifier is used with preprocessed samples. For Web track, we have submitted results of both two subtasks. Different factors are considered to improve the overall performance of our Web systems. This paper describes our methods in detail.

Keywords: TREC-11, Filtering, Web track

1. Introduction

CAS-ICT took part in the TREC conference for the second time this year, and we have submitted results of filtering track and Web track.

For filtering track, we undertook all three subtasks. Our adaptive filtering system is still based on VSM. Our Rocchio-like profile adaptation algorithm puts stress on the undetermined documents and some strategies are proposed for T11U or T11F optimization. Four runs have been submitted for evaluation: all of them are optimized for T11U measure, but in three of them T11F measure is also considered at the same time. In batch filtering and routing, we use a centroid-based classifier with preprocessed samples. Two batch filtering runs and two routing runs have been submitted for evaluation. In all of our filtering experiments, we do not use any other resources except the New Reuters Corpus.

For Web track, we undertook both the Named Page Finding task and the Topic Distillation task. Our system is based on SMART(*ftp://ftp.cs.cornell.edu/pub/smart*). In the former task, we try to integrate different factors to improve the overall system performance. In the latter task, a variant HITS algorithm is used to the top *n* results returned by SMART. Five Named Page Finding results and three Topic Distillation results have been submitted for evaluation.

2. Filtering

For filtering track, we undertook all three subtasks, but we paid more attention to the adaptive filtering task. Batch filtering and routing tasks are used to test our new classifier.

2.1 Adaptive Filtering

2.1.1 Introduction

The total 100 topics used in the filtering task this year can be divided into two sets: the first 50(R101-R150) topics are called assessor topics, which are hand-built by NIST assessors, and the last 50(R151-R200) topics are called intersection topics, which are derived from Reuters category

intersections. The two sets have been evaluated separately.

New Reuters Corpus (http://about.reuters.com/researchandstandards/corpus/) is still used this year, but the training set and testing set are different with TREC-10. The first 83,650 documents are used for training (training set) and the remaining about 720,000 documents for testing (testing set). The official adaptive filtering measures are utility (T11U, scaled using Ault's formula), and F-beta (T11F, beta = 0.5). The former is a linear utility measure and the latter is a kind of F-measure. Additionally, set precision and set recall measures are also reported in the final results. In the adaptive subtask, only three positive samples in training set are given for each topic, and the goal is to retrieve relevant documents one by one from the coming testing documents stream and get maximum T11U or T11F value at the same time.

2.1.2 System Description

Last year, we have built an adaptive filtering system, which consists of two components: the profile initialization component and the profile adaptation. This year we made some improvement based on this system, in particular, in the profile initialization and optimization modules.

2.1.3 Initialization

Our initialization process includes common operations such as term tokenization, stop words elimination, stemming, *TF* and *IDF* computation. Each topic is treated as a document and processed in the same way. The initial profile vector can be obtained by summing up the topic vector and the three positive documents vectors with different weight. Meanwhile, we set the initial threshold by computing the similarities between the initial profile and all the documents in the training set.

Since we can't use the *IDF* statistics of testing set till now, we take the *IDF* statistics of the training set as an alternative for term weighting. Ideally, we should update the *IDF* statistics when retrieving new documents from the testing documents stream. But our previous experiments have indicated that it does not seem to improve the overall filtering performance. Therefore, we use the *IDF* statistics of the training set without any modification all over our experiments.

Term selection

Last year, we applied a new method for feature selection, which can be regarded as a variation of Mutual Information. The final results indicated that our method is successful when the topic is a single Reuters category.

However, each topic of this year has been changed into a natural language statement or an intersection of some Reuters categories. Our experiment shows that the method does not work well this year. Several experiments show that the simple term selection according to the *TF* and *DF* values is a good choice.

Profile initialization

For each topic, the profile vector (denoted as $\vec{P}$) is the weighted sum of the topic vector (denoted as $\vec{T}$) and the feature vector (denoted as $\vec{F}$), which is the sum of the initial three positive documents vectors. The formula is:

$$\vec{P} = \alpha * \vec{F} + \beta * \vec{T} \qquad (2.1)$$

In our experiment, we set $\alpha=1,\beta=3$ to give prominence to the topic vector.

Similarity computation

We still use the vector *cosine* distance to compute the similarity between a profile vector ($\vec{P}_i$)

and a document vector($\vec{D}_j$). *TFIDF* value is used in our system, which is computed by $(\log(TF_i)+1)*\log(1+\frac{N}{DF_i})$, where *N* is the number of the total documents in the training set.

2.1.4 Adaptation

For each topic, after initializing the profile and the threshold, we can scan documents one by one from the testing set. If the similarity between the profile and the document is higher than the threshold, the document is retrieved, else not. Then we check the answer list of the testing set to find whether the document is really relevant or not. With this information, we can take some kind of adaptation to improve the system performance. The adaptation may include threshold updating and profile updating.

Threshold adaptation

As we know, the goal of the TREC-11 adaptive filtering system is to get maximum T11U or T11F. Therefore, we adjust the threshold for T11U optimization or T11F optimization.

For T11U, our direct goal is to avoid negative utility value for each topic. When the utility value becomes negative during filtering, which means the system retrieves too many non-relevant documents, we augment the threshold to reduce the number of retrieved documents. Another optimization strategy we take is to improve the precision while the recall can't be greatly reduced.

For T11F, our goal is to avoid retrieving zero "relevant" documents. We reduce the threshold when the system retrieves zero documents at an interval.

Profile adaptation

As the filtering task indicates, each profile vector represents a user's interest. After retrieving more and more relevant or non-relevant documents, we can get more and more useful information about the user's interest, which can help us adapt the profile. Our profile adaptation includes positive adaptation, negative adaptation and adaptation based on undetermined documents. For positive adaptation, we add the positive documents vectors to the old profile vector with weight α. For negative adaptation, we subtract the negative documents vectors from the old profile vector with weight *β*. For adaptation based on undetermined documents, we set a relative high threshold (we use *t*=0.6) to filter the retrieved undetermined documents. Those retrieved documents that have similarity below *t* are regarded as pseudo-negative documents and treated as real negative documents. A pseudo-negative document is used in negative with smaller β value. When retrieving the *n+1* th document D_{n+1}, we can adapt the *n*th profile to the *n+1* th profile according to the following formula:

$$\vec{P}_{n+1}=\begin{cases}\vec{P}_n+\alpha*\vec{D}_{n+1} & \text{if } D_{n+1} \text{ is relevant}\\ \vec{P}_n-\beta*\vec{D}_{n+1} & \text{if } D_{n+1} \text{ is irrelevant}\\ \vec{P}_n-\beta'*\vec{D}_{n+1} & \text{otherwise and } \mathrm{sim}(\vec{P}_n,\vec{D}_{n+1})<t\end{cases} \tag{2.2}$$

Thus after we have retrieved *n+1* documents, all the retrieved documents are divided into four sets: the relevant set denoted as $\{D^+\}$, the irrelevant set $\{D^-\}$, the undetermined but pseudo-negative set $\{D_u^-\}$ and the remaining documents set $\{D_u^+\}$. We do not use $\{D_u^+\}$ in the adaptation. Then the new profile vector is computed by：

$$\vec{P}_{n+1}=\vec{P}_0+\alpha*\sum_{D_i\in\{D^+\}}\vec{D}_i-\beta*\sum_{D_i\in\{D^-\}}\vec{D}_j-\beta'*\sum_{D_k\in\{D_u^-\}}\vec{D}_k \tag{2.3}$$

Formula (2.3) is some kind of the Rocchio[15] algorithm except one point: we do not compute the centroid of a document set and regard all documents in each set as one vector. In other words, we emphasize the retrieved documents and endow them the ability to adjust the profile vector quickly. As in last year, we investigate the values of α, β and β'. In our experiments, we set α=1, β=1.8 and β'=1.3.

Undetermined documents processing

In TREC-11, the relevance of most documents in the testing set is unknown to the system. In order to get more feedback information, we make some experiments on the undetermined documents.

Experiment 1: Ignoring the undetermined documents when filtering, we adjust the threshold only according to the relative proportion between the known relevant documents and irrelevant ones. But there is an important presupposition that such a distribution is the same in the undetermined documents. Unfortunately, we can' t prove this presupposition.

Experiment 2: A simple idea is that if we could know the real relevance of all documents in the testing set, the adaptation strategy proved effective in TREC-10 can still be applied. Therefore, we make a positive centroid and a negative centroid with the retrieved relevant and irrelevant documents during retrieving the testing set. When retrieving an undetermined document, we judge its relevance by computing its distance from the positive centroid and the negative centroid. Those undetermined documents that are nearer to the positive centroid will be treated as real relevant documents, while others will be treated as irrelevant documents. Thus we can simulate a situation as in TREC-10. This method allows the system retrieving plenty of "relevant" documents, which is helpful to the recall but against the precision. It seems that the initial values of the positive centroid and negative centroid greatly affect the judgment of undetermined documents. The positive centroid can be made by the known three positive samples, but we can't make a good negative centroid because we haven't any negative samples.

Experiment 3: Suppose the answer list has provided most real relevant documents in the testing set, we treat all or most of the undetermined documents as irrelevant documents. As we've introduced above, a threshold t can be used to filter the undetermined documents, those have similarity below t will be treated as irrelevant documents. The discussion of TREC-11 filtering mailing list shows that such a supposition is reasonable. With this method, we can control the retrieved "relevant" documents effectively, which is helpful to the precision. But when the number of real relevant documents in the testing set is big, such a system will suffer a heavy loss.

Of the three methods above, we apply the third one finally, partly suggested by the discussion of TREC-11 filtering mailing list. The results are encouraging.

2.1.5 Evaluation Results and Analysis

We have submitted four adaptive filtering runs: all for T11U optimization, in three of them we make balance between T11U and T11F. ICTAdaFT11Ud is optimized for recall, avoiding the heavy loss of relevant documents. As to the optimization method, we use local maximum optimization strategy at every adaptation interval to obtain the holistic maximum. We also adopt a method to avoid zero return at next interval by learning from the current adaptation interval.

Table 2.1 shows the results of the 50 assessor topics. Table 2.2 shows the results of the 50 intersection topics. Table 2.3 is the evaluation results of all 100 topics. Of the assessor topics, the system exhibits a good performance. But of the intersection topics, the system behaves badly.

Run ID	MeanT11U	T11U vs. median(topic nums)			MeanT11F	T11F vs. median(topic nums)		
		>(Best)	=	<(Worst/Zero)		>(Best)	=	<(Worst/Zero)
ICTAdaFT11Ua	0.475	46(6)	3	1(0/0)	0.427	43(5)	0	7(2/2)
ICTAdaFT11Ub	0.475	46(6)	3	1(0/0)	0.428	43(5)	0	7(2/2)
ICTAdaFT11Uc	0.471	45(6)	3	2(0/0)	0.422	41(4)	0	9(2/2)
ICTAdaFT11Fd	0.321	18(0)	2	30(3/3)	0.306	29(0)	2	19(2/2)

Table 2.1 ICT adaptive filtering runs(Assessor topics) in TREC-11

Run ID	MeanT11U	T11U vs. median(topic nums)			MeanT11F	T11F vs. median(topic nums)		
		>(Best)	=	<(Worst/Zero)		>(Best)	=	<(Worst/Zero)
ICTAdaFT11Ua	0.335	50(18)	0	0(0/0)	0.061	12(5)	32	6(6/6)
ICTAdaFT11Ub	0.330	49(17)	0	1(1/1)	0.062	13(3)	31	6(6/6)
ICTAdaFT11Uc	0.335	50(18)	0	0(0/0)	0.061	12(5)	32	6(6/6)
ICTAdaFT11Fd	0.240	19(0)	7	24(3/3)	0.052	21(1)	24	5(5/5)

Table 2.2 ICT adaptive filtering runs(Intersection topics) in TREC-11

Run ID	MeanT11U	T11U vs. median(topic nums)			MeanT11F	T11F vs. median(topic nums)		
		>(Best)	=	<(Worst/Zero)		>(Best)	=	<(Worst/Zero)
ICTAdaFT11Ua	0.405	96(24)	3	1(0/0)	0.244	55(10)	32	13(8/8)
ICTAdaFT11Ub	0.4025	95(23)	3	2(1/1)	0.245	56(8)	31	13(8/8)
ICTAdaFT11Uc	0.403	95(24)	3	2(0/0)	0.2415	53(9)	32	15(8/8)
ICTAdaFT11Fd	0.2805	37(0)	9	54(6/6)	0.179	50(1)	26	24(7/7)

Table 2.3 ICT adaptive filtering runs(all 100 topics) in TREC-11

We had partly noticed the problem of intersection topic in our experiment. It seems that the intersection topic itself makes the VSM unsuccessful. After comparing the assessor topics with the intersection topics, we guess the reason maybe that the natural language style of the assessor topics makes them appropriate to be represented and computed with vectors, while the intersection topics are not, because the different dimensions of an intersection topic vector have no internal relations as organic as those of a natural document. Another reason we guess is that there are few relevant documents on each topic in the testing set that can be used to adjust the profile vector. In TREC-10 our system has proved suitable for “big” topics but not so for “small” topics. The results of last year have also proved that as long as enough relevant documents can be provided, on the intersection-like topics we can still obtain good performance. Although in such circumstances we may not make a good initial profile vector, enough feedback can greatly adapt it to the best position. But this year the case is different. We don’t have so many relevant documents, so the weakness of VSM on the intersection topics becomes distinct. An evidence is that our system still gets better scores on most intersection topics with relative more relevant documents, such as topic R164, R175, R185, R186 and R199.

In next step, our goal is to find a new way to effectively process the semi-automatically made intersection topics. We believe such topics represent the trend in future and are worthy of much more efforts. Accomplishment of the efforts will to some extent lighten assessors’ burden in the filtering task.

2.2 Batch Filtering and Routing Subtasks

2.2.1 Text Representation

In our batch and routing filtering system, when preprocessing the documents, we give additional prominence to the words that occur in the <*title*> field and we only use *TF* weight in the vector representation.

2.2.2 Samples Preprocessing

We believe some samples in the training set are not good enough to train the classifier, so we want to eliminate them beforehand. Indeed, samples have different weights since features of documents have different weights. Importance of samples and importance of features are closely related:

- An important sample contains many important features;
- An important feature appears in many important samples;

We calculate the weights of samples as following:

Let A_{mn} is the matrix of the feature frequency in each sample, m is the number of the documents and n the number of the features. a_{ij} is the frequency of the jth feature in the ith sample.

The weight vectors of samples and features are respectively $W_f = (W_{f_1}, W_{f_2}, \cdots, W_{f_m})'$ $W_t = (W_{t_1}, W_{t_2}, \cdots, W_{t_n})'$. Their initial values are $W_f^{(0)}$ and $W_t^{(0)}$, with each component set to 1.

The formulas below are to compute the weights. It can be proved that the computing process is convergent.

$$W_{t_j}^{(k+1)} = \sum_{i=1}^{m} A_{ij} * W_{f_i}^{(k)} \tag{2.4}$$

$$W_{f_i}^{(k+1)} = \sum_{j=1}^{n} A_{ij} * W_{t_j}^{(k+1)} \tag{2.5}$$

$$(j = 1, 2, \ldots, n,\ \ i = 1, 2, \ldots, m)$$

After computing the weights of all samples, for each topic, we remove the lowest 10% samples and use the remaining samples to train the classifier.

2.2.3 Training

The system uses Rocchio method in the training process. For topic i, its representative feature vector $\vec{P}_i$ is calculated as following:

$$\vec{P}_i = \vec{P}_+ - \beta \vec{P}_- \tag{2.6}$$

Where $\vec{P}_+$ is the centroid of the relevant documents and $\vec{P}_-$ is the centroid of the irrelevant documents in the training set, β is an experiential parameter.

Since the file *filter2002_qrels.test* cannot be used for training, we use the training set to choose proper values of β and the threshold by LOOCV (*Leave-one-out cross-validation*), which is the most extreme and most accurate version of cross-validation.

In test process, those documents with high *cosine* distance to $\vec{P}_i$ are retrieved to form the final results.

2.2.4 Evaluation Results and Analysis

We have submitted two batch-filtering runs and two routing runs. All of them are optimized for T11U. The only difference between the two runs are thresholds and the parameter β in the formula (2.6).

Batch Filtering

The evaluations of batch results are shown in Table 2.4. Table 2.4 shows that in each run, the scores of T11U and T11F are close to medians. For the first 50 topics, we get a set precision higher than the median, but the set recall is lower than it. For the last 50 topics, we set a very strict threshold to avoid T11U becoming negative, because the baseline of T11U is 0.333. As a result, the scores of T11F, Set Precision, and Set Recall are all very low. Since we have set the same threshold for all 100 topics, we think the results show that the threshold for every topic should be different.

Run ID	T11U	Median T11U	T11U vs. median (Topic nums)			T11F	Median T11F	T11F vs. median (Topic nums)		
			>	=	<			>	=	<
ICTBatFT11Ua(1-50)	0.35	0.377	20	10	20	0.18	0.233	19	5	26
ICTBatFT11Ub(1-50)	0.323		15	7	28	0.248		26	7	17
ICTBatFT11Ua(51-100)	0.333	0.254	47	2	1	0	0.024	0	17	33
ICTBatFT11Ub(51-100)	0.304		40	4	6	0.011		4	17	29

Table 2.4 ICT batch filtering runs (all 100 topics) in TREC-11

Routing

We set a lower threshold to get 1000 documents for each topic to form the routing results. The only difference between the two runs is the parameter β.

Run ID	Average precision	Average precision vs. medians(Topic nums)			All Results		
		>	=	<	min	med	max
ICTRouFT11Ua(1-50)	0.243	26	7	17	0	0.223	0.507
ICTRouFT11Ub(1-50)	0.25	31	8	11			
ICTRouFT11Ua(51-100)	0.024	18	16	16	0	0.02	0.085
ICTRouFT11Ub(51-100)	0.025	18	18	14			

Table 2.5 ICT routing runs (all 100 topics) in TREC-11

We can see that all of our results are similar to the medians. We think this is because we only set one same threshold for all topics and lack an effective parameter optimization method. We will try to research on automatic parameter optimization methods.

In the future, we have a lot of work to do to improve our work. For feature selection, we want to use N-Gram to add more terms to represent the documents. For the last 50 topics, we have tried to use KNN to improve the classification results. To our surprise, its result is much worse than the Rocchio method. We will research on the phenomenon and try more complex methods.

3. Web Track

3.1 Introduction

Last year we took part in TREC for the first time and we only submitted four runs for the ad hoc task. This year we submitted runs for both two tasks.

This year, Web track consists of two new subtasks: the Named Page Finding task, which is introduced to investigate methods for finding a particular page that has been named by the user, and the Topic Distillation task, which is introduced to investigate methods for finding key resources in a particular topic area. In the former task, the system should return a single named page as the result. For instance, for the query *"passport application form"*, the correct answer

should be the page *travel.state.gov/dsp11.pdf*, which contains the electronic copy of requested form. In the Topic Distillation task, a single relevant document is not important any more. The concept *resource* is introduced as the basic element of results and judgments. The test collection of this year's Web track is changed to *.Gov* data set which substitutes *Wt10g* used in previous years.

Though the Web track tasks have been significantly modified, the basis of experiments is still the traditional IR systems. In TREC 2001 we investigated the effectiveness of the combination of classical Boolean model and probabilistic model in the ad hoc task. We also investigated methods that make use of link information between pages in the same task. Neither of the results was as good as we had expected. So this year we decide to adopt vector space model and to make use of only text contents and internal structure of pages. Our retrieval system is based on SMART. In order to deal with large data set such as *Wt10g* and *.Gov* test collection, we modified the basic SMART system, and the *Lnu-Ltu* weighting method was added to the system. This method has been proven to be very effective and efficient in our experiments. The classical weighting methods such as *lnc-ltc* do not behave well in our experiments.

3.2 Named Page Finding Task

As introduced above, the goal of Named Page Finding task is to find appropriate page(s) named by users. It is rather close to a special kind of user requirement, i.e., finding a few documents that precisely meet the information need of users. The query *"passport application form"* is an example. Another one is the query "table of contents gnu make manual", by which a user would like to find the exact page that is the table of contents of GNU make manual. By analyzing these examples we have found some features that can be utilized.

Firstly, the content-based ranking score of traditional IR system is still the most important factor in Named Page Finding. If we assign the content-based score a less important coefficient in result merging process that will be described below, the final results will be worse. This can be explained if we notice that single term is more important in Named Page Finding task than in ad hoc task. This task pays more attention to precision than to recall. Only those pages that contain all or most of the query terms would have high possibility of meeting information need implied by the query in, thus they would have higher content-based scores than most of the irrelevant documents. Certainly some of irrelevant documents will also have high content-based scores, but we will enhance the scores of relevant documents by result merging process.

Secondly, the internal structure of documents will give us plenty of information. As the name of task suggests, query terms of Named Page Finding task are the names of relevant documents. Usually they are precise representations of topics. They should more possibly appear in important positions such as document title, beginning sentences of paragraphs and section headers, or display in a striking manner, for example a bold, italic, and large size font face. In such situation authors of documents have explicitly defined them as important terms. We can get a lot of relevance information by comparing the query terms with them. Besides this there is another reason why the method is especially useful for the task. Queries in ad hoc task are often about general topics. They must be described by natural language so that people can understand the information need under which the queries are developed. So they are prone to ambiguity. Correspondingly the relevant documents cannot be named clearly and easily. On the contrary, the information need of Named Page Finding task can be very easily understood, even without extra descriptions, so authors and searchers of the same documents will in the gross adopt the same terms as topic descriptions. The Homepage Finding task in last year's web track can be regarded

as a kind of Named Page Finding task. In fact, when we added the phrase "home page" to the original queries we got obvious improved results. In our contrast experiment, ad hoc runs using document structure information gave poor results whose average precisions are too low to be mentioned while Homepage Finding runs gave fairly satisfactory results.

The last factor we have proven to be effective for the Named Page Finding task is anchor texts of documents. They act as almost the same role as the second factor. They can be regarded as names given by referrers to target documents. When the target documents can be easily named and referrers adopt the same names widely, retrieval results using the names are fairly satisfactory.

As we have stated above we believe that Homepage Finding task is a special kind of Named Page Finding task. So except some special methods for Homepage Finding such as analysis of URL depth, the methods that are effective for Homepage Finding should also be effective for Named Page Finding. We ran our experiments on *Wt10g* data set using topics and qrels developed for the Homepage Finding task to find the most optimized parameters. The results are shown in Table 3.1 and Table 3.2. We then applied the same system to the .*Gov* data set and Named Page Finding task. The experimental results that we observed have proven to be satisfactory.

We use the linear result merging method to get the last result of Named Page Finding task. The merging formula is

$$W(p) = \alpha * w_c(p) + \beta * w_s(p) + \gamma * w_a(p) \quad (3.1)$$

Where $w_c(p)$ is the content weight of page p, $w_s(p)$ is the weight from structure information, $w_a(p)$ is the weight from the anchor text of page p and α,β,γ are their coefficients. In our experiments only the titles of documents are used as structure information. The evaluation results are shown in Table 3.3.

Average Precision	R-precision	Recall
0.1938	0.2185	2243

Table 3.1 Our content-based experiment for the ad hoc task of TREC-10.

Content(α)	Structure(β)	Anchor text(γ)	MRR	Correct Answers
1	0	0	0.4185	122/145
0	1	0	0.4467	105/145
0	0	1	0.3769	94/145
1	0.5	0.5	0.5880	133/145
1	0.5	0.8	0.6032	130/145
1	0.5	1	0.5806	130/145

Table 3.2 Our Homepage Finding experiments of TREC-10

Run ID	MRR	Answers Found@10	Not Found@all
ictnp2	0.559	114/150	18/150
ictnp3	0.557	116/150	18/150
ictnp4	0.555	116/150	18/150
ictnp6	0.613	127/150	14/150
ictnp7	0.613	127/150	14/150

Table 3.3 ICT Named Page Finding runs in TREC-11

3.3 Topic Distillation Task

As described in the TREC-2002 Web Track Guideline, a key resource might be:

- The home page of a site dedicated to the topic.
- The main page of a sub-site (part of a site) dedicated to the topic. (If there are several relevant pages but no main page linking them, then the individual pages must be judged on their own merit.)
- A highly useful html, pdf, doc, ps page dedicated to the topic (should be an outstandingly useful page). Return the page's URL.
- A highly useful page of links (hub page) on the topic. Return its URL.
- A relevant service e.g. perhaps http://www.nasa.gov/search/ for the NASA topic.

Except the last two cases key resources are some important pages inside individual sites. Our first experiment was based on HITS algorithm. We submitted queries to SMART and retrieved ranked page lists, and then applied HITS to every group of pages coming from the same site. We extracted the page that had the maximum Hub+Authority value from each group of pages and added them to the final result. We found that the average result of this method was disappointing, partly because many Hub and Authority pages computed by HITS cannot meet the definition of key resource. Our last experiment on this task was based on a simple idea. After the first retrieval, we scanned the page list. If we found a page's url containing the other's, we then re-weighted the latter page by adding the former's weight to the latter's. After re-weighting the weight of a certain result page x is

$$w = \sum_{p} \frac{w_p}{\sqrt{r_p}} \tag{3.2}$$

Where p is a page whose url string contains x's, w_p is the content weight of page p and r_p is the rank of page p. The run icttd2 is based on this approach, and ictted3 is based on icttd2 plus some additional re-weighting methods. The evaluation result is shown in Table 3.4.

The run icttd1 is a baseline run produced by our retrieval system. It is the best one among the three runs. It seems that our re-weighting methods are not so effective as we have expected. We believe that more attentions should be paid to the instances of key resources given by the TREC qrels so that characters of them can be found.

Run ID	Average Precision	R-Precision	Rel_ret
icttd1	0.1620	0.1919	1038/1574
icttd2	0.1364	0.1599	1038/1574
icttd3	0.0597	0.1034	288/1574

Table 3.4 Result of Topic Distillation task in TREC-11

4. Conclusion

We've participated in the TREC conference for two times. By communicating with the researcher all over the world, we've learned more. We've got many experiences in English information processing, which will benefit us greatly in our Chinese information processing.

TREC not only advances our research on IR, but also enlighten our insights. From here, we can find our advantages and disadvantages comparison to the foreign friends going the same way. We are glad to take part in TREC continuously.

Acknowledgements

This research is supported by the national 973 fundamental research program under contact of G1998030413, the Institute Youth Fund under contact 20016280-9 and the Institute Youth Fund under contact 20026180-24. We give our thanks to all the people who have contributed to this research and development, in particular Yanbo Han, Li Guo, Qun Liu, Xin Zhang, Hao Zhang, Dongbo Bu and Huaping Zhang.

References

[1] Ogawa, Y., Mano, H., Narita, M., Honma, S. Structuring and Expanding Queries in the Probabilistic Model. In *The Ninth Text REtrieval Conference (TREC 9),2000.*

[2] O. Yasushi, M. Hiroko, N. Masumi, H. Sakiko. Structuring and expanding queries in the probabilistic model. In *The Eighth Text REtrieval Conference (TREC 8),1999.*

[3] S.E. Robertson, S. Walker. Okapi/Keenbow at TREC-8. In *The Eighth Text REtrieval Conference (TREC 8),1999.*

[4] S.Brin and L.Page. The anatomy of a large scale hypertextual web search engine. In *The 7th WWW Conference,1998.*

[5] Lawrence Page,Sergey Brin,Rajeev Motwani,Terry Windograd. The Pagerank citation ranking: Bring order to the web. *Stanford Digital Libraries working paper, 1997-0072.*

[6] J.Kleinberg. Authoritative sources in a hyperlinked environment. *Proc 9th ACM-SIAM SODA,1998.*

[7] Ian H. Witten, Alistair Moffat, Timothy C. Bell. *Managing gigabytes: Compressing and indexing documents and images, 2nd ed, 1994.*

[8] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford. OKAPI at TREC-3, In *The Third Text REtrieval Conference (TREC 3),1994*.

[9] S. Robertson, I. Soboroff, The TREC 2001 Filtering Track Report, In *The Tenth Text REtrieval Conference (TREC 10)*, page 26, 2001.

[10] B. Wang, H. Xu, Z. Yang, Y. Liu, X. Cheng, D. Bu, S. Bai, TREC-10 Experiments at CAS-ICT: Filtering, Web and QA, In *The Tenth Text REtrieval Conference (TREC 10)*, page 109, 2001.

[11] Yi Zhang, James P. Callan, Maximum Likelihood Estimation for Filtering Thresholds, SIGIR 2001, page 294-302, 2001.

[12] Y. Zhang, J. Callan, The Bias Problem and Language Models in Adaptive Filtering, In *The Tenth Text REtrieval Conference (TREC 10),* page 78, 2001.

[13] T. Ault, Y. Yang, kNN, Rocchio and Metrics for Information Filtering at TREC-10, In *The Tenth Text REtrieval Conference (TREC 10)*, page 84, 2001.

[14] S. Alpha, P. Dixon, C. Liao, C. Yang, Oracle at TREC 10: Filtering and Question-Answering, In *The Tenth Text REtrieval Conference (TREC 10),* page 423, 2001.

[15] Rocchio, J. J. Relevance Feedback in Information Retrieval. In *The SMART Retrieval system, Prentice-Hall, Englewood NJ. 1971, 232-241.*