THUIR at TREC 2009 Web Track: finding relevant and diverse results for large scale Web Search[1]

ZC Li, F Chen, QL Xing, JW Miao, YF Xue, T Zhu, B Zhou, RW Cen, YQ Liu, M Zhang, YJ Jin, SP Ma
State Key Lab of Intelligent Technology & Systems,
Tsinghua National Laboratory for Information Science and Technology,
Dept. of Computer Science & Technology, Tsinghua University, Beijing, 10084, China
z-m@tsinghua.edu.cn

Abstract: This is the 8th year that IR group of Tsinghua University (THUIR) participates in TREC. This year we focus on Web track, which contains two tasks, namely ad hoc and diversity. On ad hoc task, we improved the efficiency of our distributed retrieval system TMiner to handle terabytes of Web data. Then three studies have been done, namely page quality estimation, ranking feature analysis, and model comparison. On diversity task, we proposed several new approaches on searching strategy, user intention detection, and duplication elimination. To mine user's intention, we proposed and compared two different strategies, namely 'searching + content-based diversity' which is a kind of result clustering, and 'user based diverse intention prediction + searching' which is in the branch of query expansion.

1 Introduction

IR group of Tsinghua University (THUIR) participates in this year's Web track. It's also our 8th year's TREC experience.

On ad hoc task, we improved the efficiency of our distributed retrieval system TMiner to handle terabytes of Web data. Then three studies have been done, namely page quality estimation, ranking feature analysis, and model comparison.

First, we propose several features for web page spam detection, including title, length, content compression ratio, keyword-based filtering, and PageRank features. The impacts of the features are observed. Then an effective algorithm has been proposed to filter spam pages.

Second, we embedded different types of ranking features in our distributed retrieval system TMiner. One is link and web usage based page importance factors, such as PageRank, top sites based on website traffic, etc. Another is the usage of different content of the document, namely original page full text, and in-link anchor text.

Third, we compare the effectiveness of different ranking models on searching with large scale data. One is the BM25-based probabilistic model, another is the improved probabilistic model with word-pair proximity by TMiner, and the last one is the probabilistic model by Lucene.

On diversity task, we proposed several new approaches on searching strategy, user intention detection, and duplication elimination. To mine user's intention, we proposed and compared two different strategies, namely 'searching + content-based diversity' which is a kind of result clustering, and 'user based diverse intention prediction + searching' which is in the branch of query expansion.

One is to automatically find the self organized topics of content by result clustering. In this strategy, the

[1] Supported by Natural Science Foundation (60736044，60903107) and Research Fund for the Doctoral Program of Higher Education of China （20090002120005）

level and the granularity of the clusters is one of the key factors to consider. A hierarchical incremental online clustering algorithm is proposed for both effectiveness and efficiency. Furthermore, we also used a content analysis algorithm to extract the core part of the page. Then we made comparative study on the clustering result base on the extract content and the original full text. This is the strategy of 'searching + content-based diversity'.

Another one is to predict the user's intention before search. A good choice is to use user log which is hard to achieve. So an alternative choice is to use the external resource. We automatically submit each query to the commercial search engine, and collected the query suggestion and related search queries without any manual labeling. We assume that each new query represents one aspects of the potential user intention. Then each new query is taken as the expanded one and submitted to our retrieval system. To combine the multiple result lists, we implemented the diversity result selection algorithm, which selects two or three most relevant pages for each user intention based on the probability.

To make a comparative research, we also generate a baseline study which is 'search + duplicate elimination' and no efforts on finding diverse intentions. Both content-based and site-based duplicate elimination approaches are integrated.

2 Web page spam detection

On real world search task, web spam pages are considered as one of the dominant factors to hurt the search system performance. In our data cleansing work, 4 features were proposed, including title length, content length, content compression ratio and keyword-based filtering. To draw the threshold analysis, a small sample set, including 100 pages, is created according to the 4 features and manual annotation is made. Since this training set is not large enough, the parameters set in the final experiments might not be optimized. And further analysis will be made afterwards.

Title length: One popular practice of creating spam pages is "title keyword stuffing" [1]. During the process of stuffing, the title of the web page is growing larger with lots of keywords which are not relevant to the content of the page. So that the page title will match much more queries and can be accessed by more users. To find this kind of pages, we made the statistics of the title length. But unfortunately, on the rough observing results, nearly half of the pages with long title are not spam, and even high-quality one. Hence this feature is not used in the final experiment.

Content length: The underlining assumption for this feature is that a page with little words may contain nothing but spam, or at least less useful information. But the result is not encouraging based on the initial analysis.

Content compression ratio: Repeating hot query words in the content is another popular way of SEO. Hence the *content compression ratio* (CCR, or the *compressibility*) [1] will be much more different than normal pages. The content compression ratio is calculated by the following Eq. 1.

$$\text{CCR} = \text{content length (in terms of \# of words)} / \text{\# of vocabulary} \qquad \text{(Eq. 1)}$$

When the CCR is larger than θ, the page is taken as spam. According to the observation on the sampled page set, the threshold θ is set to 8.5.

Keyword-based filtering: Porn words' filtering is also one of the anti-spam techniques in real world search engines. A list of porn words was found from the internet [2]. When the numbers of the porn words in the page is larger than α, then the page is taken as the spam. In our experiments, the threshold is set to 16.

The spam filtering work is embedded into our ranking system, as shown in the following section 4.1. By

this spam pages detection, 1873 pages have been filtered in retrieval results list.

3 Data and system preparation

3.1 Training set construction

To tune the parameter of system and methods, we construct a small training set with 11 queries and corresponding answers. The query set is manually selected according to the 'hot query list' of commercial searching engines, which are different from the queries used in Web track task. Then the answer set is constructed with the pooling technique. Documents from three different sources are gathered into the pooling set. First, we submitted these queries to our TMiner system retrieved on the full text of the ClueWeb dataset (the data of the Web track task), and gained the top 50 documents for each query. Then the same system with anchor text index of the ClueWeb is used to gain the top 50 documents for each query. To avoid overtraining, we collected the top 100 results of online Google search engine for each query, and kept the documents in the ClueWeb. Then 11 assessors were asked to annotate these queries, each of them was assigned two queries and each query was annotated by two persons. Hence the final training dataset is generated, which is used to tune all the parameter of our system and algorithms.

3.2 System preparation

We improved the efficiency of our distributed retrieval system TMiner to handle terabytes of Web data. 500 million web documents are divided into 138 barrels. Each barrel is considered as a unit node of the distributed system, which can index and retrieval independently.

In the pre-processing step, the original html tags are removed, while user defined tags are added. These tags are used to label the different fields of the text, including title, Meta keyword, Meta description, bold, italic, sub header and anchor text. The anchor text is made by link analysis module. Not all the terms in the web page will be indexed. Firstly, 466 stopwords and all the punctuation except "-" are removed; secondly, terms whose length is greater than 25 and terms with digit whose length is greater than 4 are removed. Thus, much noise are Eliminated. For the mass data, stemming technique is not used. The filtered web text and anchor text are built into inverted file index. The positions in the text and field type of the term are recorded in the index item for each document. Each barrel manages the data independently; therefore, the global information such as global document frequency of terms and average document length need be calculated separately. The features of each document for spam detection such as title length, content compression rate, and keyword-based filtering, are all built into the index, as well as the document quality factor such as pagerank, which can help the retrieval. The size of each index is between 12GB and 17GB except for some especially small ones.

4 Ranking features and models

4.1 Improved probabilistic model in our TMiner system

In retrieval step, each barrel returns 1000 documents ranking according to the relativity to the query, and then the results of all the barrels are merged to generate the final document list. The traditional probabilistic modal is used to measure the relationship between the query and a document. Several additional techniques are recommended.

We use "AND search" to the query, which require the retrieved document contain all the terms of the query. The conjoint terms in the query which co-occur in the document will enhance the relativity, which we call the word pair model [3] (shown as W_{wp} in the Eq 2). Further, when a term occurs in a special field, a higher weight of the term is signed. As the pagerank and spam detection method added, finally the relevance

between query Q and document D is computed as Eq 2.

$$R(Q,D) = \left[W_{BM\,25} + \alpha_1 \cdot W_{wp} + \alpha_2 \cdot \log_{10}\left(PageRank(D)\right)\right] \cdot I(Q \subset D) \cdot I\left(\overline{spam}\right)$$

$$W_{BM\,25} = \sum_{i=1}^{m}\left(\log\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot |D| / avgdl\right)}\right), \qquad \text{(Eq 2)}$$

$$W_{wp} = \sum_{i=1}^{m-1}\left(\log\frac{N - n(q_i q_{i+1}) + 0.5}{n(q_i q_{i+1}) + 0.5} \cdot \frac{f(q_i q_{i+1},D)\cdot(k_1+1)}{f(q_i q_{i+1},D) + k_1 \cdot \left(1 - b + b \cdot |D| / avgdl\right)}\right),$$

$$f(q,D) = \sum_{i=0}^{k} \beta_i \cdot f(q, field_i)$$

Three parts are linear combined in the model, namely traditional probabilistic BM25 modal weight W_{BM25}, word pair weight W_{wp} and pagerank of the document *PageRank* (*D*). I() is indicator function, return value 1 when the event occurs, 0 otherwise. Event Q⊂D means the document D contains all the terms in query Q. Event $\overline{spam}$ means that the document is not detected as spam by the spam detection module described in the section 2. *N* is the total number of documents, *n*(*q*) is the number of documents contain *q*, k_1 and *b* are experimental parameters of BM25 ranking, |*D*| is the length of document *D*, *avgdl* is the average document length, *f* (*q*,*D*) is the virtual term frequency of *q* in *D* which is the total count of *q* in every field multiplied by the corresponding field weight β, W_{wp} is also gotten from traditional probabilistic modal with that conjoint terms considered as one term, α_1 and α_2 are the combination weights for word pair modal and pagerank.

After merging results of 138 barrels, the top 1000 documents are generated for ad hoc task. Parameters were tuned on the training set, by which we set α_1 = 0.2, α_2= 0.8, double the weight of the term frequency on title field and meta field.

4.2 Using Lucene BM25 package

We construct another retrieval system based on Lucene. Lucene uses VSM as its original ranking model. As the performance of VSM in Lucene is not so good as TMiner which used BM25 Model, we use a BM25 package for Lucene to improve the system. The package is written by Joaqu ń P érez-Iglesias and has BM25 and BM25f implementation for Lucene Java [4].

To compare the system performance, most of the data preprocessing is the same as that of our improved probabilistic model with TMiner system, such as 138 data buckets, global DF (Document Frequency) calculation, multi-field extraction.

We search the queries in 8 different fields, namely content, anchor text, title, meta keywords, black, meta data, italics and headings. The anchor text field has a very large weight of 7, and title field and meta keywords field are also important to have the weight of 3.

5 Finding diverse information

We think that diversity may be explored in three levels: *semantic diversity*, *topic diversity* and *service diversity.* The ***semantic diversity*** in IR is also taken as the expression's ambiguity problem. For example, when the user is querying about 'windows', he might denote to the windows of the house (in architecture domain), or find the information of Microsoft Windows (in computer domain). Even if the user is searching for Microsoft Windows, he might need the topics of windows new version release, download, windows update, etc, which goes to the ***topic diversity***. Furthermore, if the user's intention is the windows update, he might expect the windows update service URL, or the FAQ for the problems for windows update, which

leads to the different service homepage. It is called ***service diversity***. The diversity of the user's intention may include either of the three types of the diversity, or even all of them.

5.1 Diverse information finding strategy

To mine user's intention, we should firstly know how many diverse intentions could be. We call it the intention targeting problem. We proposed and compared two different strategies One is '*searching + diverse result clustering*', which first finds the information that is relevant to the user's need using similarity measure, and then clusters the information to generate the diverse topics. Another strategy is the '*diverse intention prediction + searching*', which first predicts the diverse intention of the user according to the user's behavior or background knowledge, and then finds the relevant information with each topic.

The original retrieval results are got by our ad hoc search task, which is implemented with word pair model and improved probabilistic model with PageRank ranking on the full text combined with in-link anchors, and taken spam pages detection into consideration.

5.2 Searching + diverse result clustering

5.2.1 Clustering algorithm

With '*searching + content-based diversity*' strategy, we try to automatically find the self organized topics of content by result clustering. In this strategy, a hierarchical incremental online clustering algorithm is used for both effectiveness and efficiency [5]. In this algorithm, a good way to compute similarity (distance) between every two documents is a critical factor for good results. In our method, the similarity of document d and d' is taken as following Eq 3:

$$similarity(d,d') = \sum_{w \in d \cap d'} weight(d,w) * weight(d',w) \qquad \text{(Eq.3)}$$

$$weight(d,w) = \frac{tf(d,w)\log((W+1)/wf(w)+0.5))}{\sqrt{\sum_{w' \in d}(tf(d,w')\log((W+1)/(wf(w')+0.5)))^2}} \qquad wf(w) = \sum_{d \in D} tf(d,w)$$

where *w* is any non-stop word contained in a document, function $weight$ implements the TF-IWF model. In this equation, $tf(d,w)$ stands for the frequency of word w appeared in document d, W is the frequency of all the non-stop words appeared in d, and $wf(w)$ is the frequency of word w appeared in all the four thousand documents D.

Clustering is constructed to get a new document when the similarity of any two documents is bigger than a predefined threshold. The same one of 225 was set as that in [5]. And then similarities related to this new document are refreshed in the distance matrix. Clusters are done until all similarities between any two documents are smaller than the predefined threshold.

5.2.2 Diverse result selection algorithm

After clustering, categories are obtained for a certain query. They are taken as the probable user intentions for this query, and then the improved diversifying search algorithm is adopted for people's diversified search.

The original diversity calculation model of IA-SELECT algorithm proposed by Agrawal et al [6] is showed as follows:

$$P(S \mid q) = \sum_{c} P(c \mid q)(1 - \prod_{d \in S}(1 - V(d \mid q,c)) \qquad \text{(Eq. 4)}$$

Where *S* is the final returned page list and *q* is a query dealt in this issue. *P*(*c*|*q*) stands for the probability of

query q belonging to category c, and $V(d|q,c)$ denotes the quality value of document d for query q when user's intended category is c. And the task is to choose a set S so that $P(S|q)$ reaches to its max value, which in details is the main work of [6].

Then we made some modification in our experiment:

First, we selected the top 1 document from each category by default, and made them the top $|C|$ documents ordered by descending $V(d|q,c)$,where $|C|$ is the total number of categories. There are two reasons for this. (1) We believe that top results should cover as much user intention as possible. But if one aspect of intention has a low proportion, saying $P(c|q)$, IA-SELECT algorithm may not show any document of this intention in the front. (2) Some category's top document may has a extra large value of $V(d|q,c)$ than the other documents in the category. Once this top document is selected, it will cause a big penalty to the category. For example, the documents in the category whose top document has $V(d|q,c)=1$ even never get opportunity to be selected by making $U(c|q,S)=0$ once the top document is selected, according to IA-SELECT. Hence picking out top documents in each category is one way to smooth the penalty and avoid the noise. And by this way, it's guaranteed that at least one page is chosen from every category.

Furthermore, the remains are deciding the parameters in the equations introduced above. We get these values as follows:

$$P(c \mid q) = N_c / N \qquad \text{(Eq. 5)}$$

N_c is the number of documents contained in class c after clustering, and N is the number of all the clustered documents.

$$V(d \mid q,c) = \frac{R(q,d_c) - R_{\min}}{R_{\max} - R_{\min}} \qquad \text{(Eq. 6)}$$

$R(q,d_c)$ is the score of document d in category c retrieved with query q. $R_{\min}$, $R_{\max}$ stands for $\min\{R(q,d_c)\}$ and $\max\{R(q,d_c)\}$ respectively.

Finally, although the algorithm showed above can deal with diversified intentions, more than two web pages may be chosen, which are both in the same site and the same clustered category. However, web pages from the same site and are also clustered into the same category, are more likely to be in the same subtopic. Hence domain-level duplicate elimination of the web page is performed. At last, all the documents chosen by the diversity algorithm are sorted depending on their original relevance ranking scores.

5.2.3 Web page content extraction

Besides the result clustering based on the original web pages, we also studied the clustering effect based on the extracted web page content. Hence a content analysis algorithm is used as preprocessing. In this process, the title and body of the web page are extracted, while other things, such as advertisements, content-related anchors (which we think is the description of the linked target page but not the current page), site navigations etc. are abandoned.

This task is taken without any manual labeling, so noise and incorrect filtering occur. For example, if the title inside a page is set by script code rather than text, or the content is rather short with many pictures, our extracting tool cannot work effectively as expected.

Finally, experiments are done to make compares between these two methods (clustering based on the original web page, shown as method 1; and clustering based on extracted content, shown as method 2). Take the first query "obama family tree" for example, after clustering, 422 categories are left in the first method, and 460 categories in the second. By the way, the final results returned to users are 1000, so about 2.5 web pages should be chosen from every category on average. From these experiments, it's found that

for a certain query, documents laid in the first half of the returned list are almost the same with both methods, except few documents appear only in one list. But in the latter half, there exists many differences in both documents and their ranks.

5.2.3 More discuss on the clusters control

The following table shows the effectiveness by using different number of the clusters in our experiments. The results show that different control degree of the number of the clusters do not affect the result much.

Strategy	**Selection alg.**	**#Clusters**	**alpha-nDCG@10**	**IA-P@10**
1	improved IA-SELECT	445	0.234	0.094
1	cluster-based direct selection	445	0.237	0.097
1	improved IA-SELECT	48	0.235	0.097
1	cluster-based direct selection	48	0.233	0.096

5.3 Diverse intention prediction + searching

The strategy of '*user based diverse intention prediction + searching*' is to predict the user's intention before search. A good choice is to use user log but large scale real data is hard to achieve. So an alternative choice is to use the external resource.

First, in order to obtain sub-topics of each query, we automatically submit each query to Google, and collected the query suggestion and related search queries without any manual labeling. The query suggestion refers to the queries in the prompt box while typing the query to the query box, and related search queries refer to the recommended queries shown at the bottom of the result page.

Second, some rules is conducted to filter the queries from Google and obtain more relevant and accurate phrases and sub-topics.

(1) The query suggestion and related search queries should fully contain the original query terms.
(2) If rule (1) is not satisfied, and the original query is a substring of one expanded query term, then the expanded query is preserved only if it is a URL like string, i.e. ended with ".com", ".org", ".edu", ".net" etc.
(3) If there are duplicate phrases satisfy the conditions above in both query suggestion and related search queries, we preserve only one.

Third, each new query is taken as the expanded one and submitted to our retrieval system.

Finally, to combine the multiple result lists, the same diversity result selection algorithm is performed as shown in the previous section 5.2.2, which selects two or three most relevant pages for each user intention.

5.4 Searching + Duplication elimination

In our duplicate eliminating study, content-based and site-based approaches are integrated.

We first calculate the cosine similarity between two document pairs, and obtain an upper triangular matrix *Aij* (where each element a_{ij} represent the similarity between document i and j, where $i < j$) as similarity matrix. Then if a_{ij} is greater than θ, the document j is eliminated. In our experiments, θ is set to 0.4.

Then in the second traversal process, site-based approach is taken as follows:

(1) We keep at most ***m*** results from a distinct website in the result list, where m is set to 2.
(2) We keep at most **one** result from a distinct website in any ***w*** coterminous results, where w is set to 5.

After the two steps, final result is generated.

6 Submitted results

We submitted three runs for each task, all of which are automatic ones. The evaluation results are shown below.

6.1 Ad hoc task (category A)

	Description	**p@5**	**p@10**
THUIR09An	TMiner system, Improved probabilistic model (with PageRank, Wordpair and anti-spam embedded), retrieved on anchor text only	0.3840	0.3740
THUIR09TxAn	TMiner system, Improved probabilistic model (with PageRank, Wordpair and anti-spam embedded), retrieved on the full page combined with anchor text	0.3800	0.3640
THUIR09LuTA	Lucene with BM25f model, re-ranking with PageRank, anti-spam, retrieved on the full page combined with anchor text	0.2120	0.2100

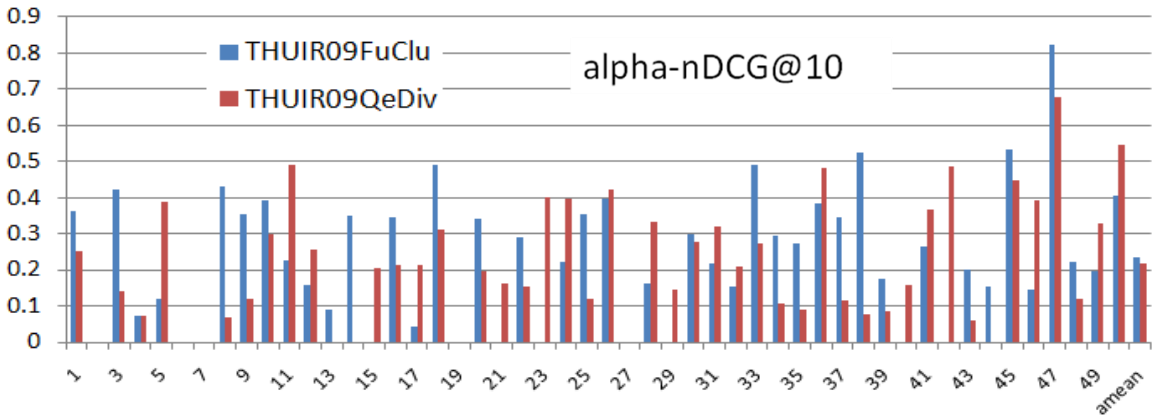
THUIR09TxAn is used for the following diversity task as the baseline result.

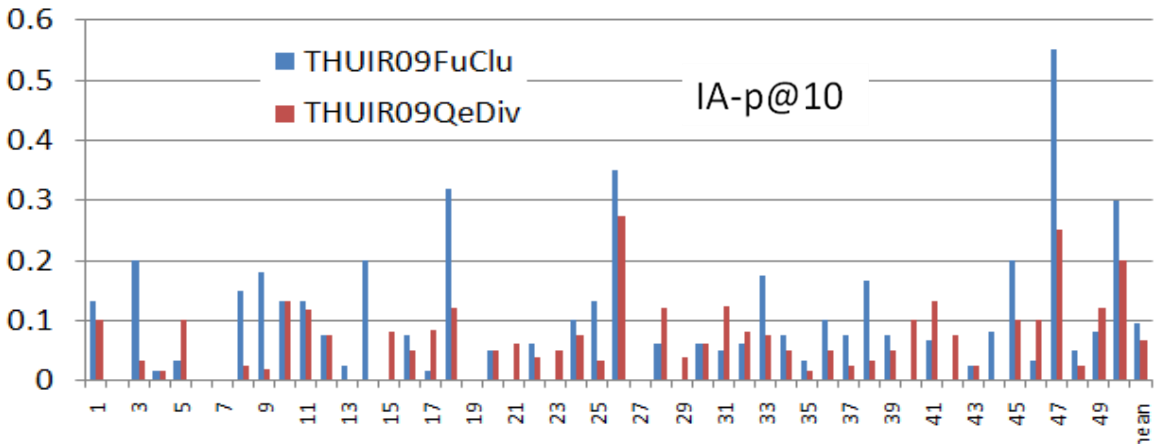
6.2 Diversity task (Category A)

Run tag	**Description**	**alpha-nDCG@10**	**IA-P@10**
THUIR09FuClu	Searching + result clustering on the full page + site-based duplicate elimination + improved IA-selection	0.234	0.094
(Not submitted)	Searching + result clustering on the full page + site-based duplicate elimination + cluster-based direct selection	0.237	0.097
THUIR09AbClu	Searching + result clustering on extracted content + site-based duplicate elimination + improved IA-selection	0.231	0.096
THUIR09QeDiv	User based diverse intention prediction (QE based on Google) + searching + improved IA-selection	0.219	0.068

6.3 Per-topic analysis

Following figures show the topic-by-topic analysis results on the two typical results of the two diversity strategies in terms of alpha-nDCG@10 and IA-p@10.





6.4 Conclusion and the discussion

This is the first year's experiment on the diversity task for Web IR. Several conclusions can be drawn by our research and there are many problems leave for further study.

First, In-link anchor text is shown to be powerful! (Even much more effective than the full text)

Second, PageRank is definitely helpful!

Third, the improved probabilistic model is effective.

Finally, on finding diverse result strategies, we conclude that (1) "Searching + Result clustering" is a good choice for diversity task; (2) the strategy of "User intention prediction + searching" still need more analysis; (3) There is bias on the task, such as the bias of the diversity definition, the bias of the diversity judgment, and bias on using different user log data to model the user's intention. In fact, these biases coming from one problem: the difficulties in understanding a global diversity, which still need further study.

Acknowledgement

We would like to thank Qian Wang, Huijia Yu, Xudong Li, Yu Sun and Wei Yang for their help on system building and data preprocessing.

References

[1] Ntoulas A,Najork M,Manasse M,et al.Detecting spam Web pages through Content Analysis, In Proc. of the 15th International Conference on World Wide Web (WWW2006), pp83-92.

[2] http://www.theporndictionary.com/view/

[3] M. Zhang, C. Lin, Y. Liu, L. Zhao, S. Ma, THUIR at TREC 2003: Novelty, Robust and Web. In the proceedings of the 12th Text Retrieval Conference, page 556-567, Maryland, US, 2003.

[4] Joaquin Perez-Iglesias, http://nlp.uned.es/~jperezi/Lucene-BM25/

[5] Chanhui Wang. Research on News Issue Construction and Topic Mining in Web Environment, Tsinghua University, 2008.10, Pages 22-25.

[6] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson. Samuel Ieong, Diversifying Search Results, In WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining, 2009, Pages 5-14.