NLPR at TREC 2004: Robust Experiments

Jin Xu, Jun Zhao, Bo Xu
National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Science
Beijing, China 100080
{ jxu, jzhao, bxu }@nlpr.ia.ac.cn

1. Overview

It is the second time that the Chinese Information Processing group of NLPR participates in TREC. In the past, we have investigated the use of two key technologies: Window-based weighting method and Semantic Tree Model for query expansion, with success, to tasks in novelty and robust tracks. We focused on the Robust Retrieval Track at this year’s conference. Based on the previous IR architecture, our research on this year's robust mainly focused on three aspects: (1) two-step retrieval scheme; (2) word sense entropy; (3) several strategies for merging multiple runs.

Our paper is organized as follows. Section 2 shows the basic architecture of our IR system and the new techniques for improving its performance. Section 2.1 presents the two-step retrieval scheme which mainly attempts to reduce the influence of noise introduced by query expansion. Section 2.2 introduces a new method for query word weighting—word sense entropy which is a measure for the variety of the sense of query word based on WordNet’s structured knowledge. Section 2.3 describes several different strategies which we have used for merging the results of multiple runs produced by different retrieval approaches. Section 3 gives the experimental verification of the techniques mentioned in section 2. Section 4 concludes our work.

2. New Techniques

Our IR system is both for English Retrieval and Chinese Retrieval. The basic architecture of the IR system and the fundamental retrieval models have been shown in the [Qianli Jin 2003]. In this year, we have experimented three new technologies for robust track.

2.1. Two-step Retrieval Scheme

As we know, query expansion methods, such as expansion based on pseudo feedback or based on semantic knowledge, usually introduce many query-irrelevant words which are called noise. It would hurt the system performance very much. Noise is one of kernel problems for the application of query expansion. In the following, we presented a two-step retrieval scheme, mainly attempting to reduce the influence of noise.

There are two characteristics for the TREC’s Robust retrieval. (1) A topic of TREC style has three fields: title, description and narrative. It can be found that, “title” field always contains *core query words* which are mostly nouns and often have discriminative function for this topic's retrieval process, while the "description" field and the "narrative" field are similar to the expansion of the "title" field, because they include more detailed information about this topic. (2) Robust does not require sorting all relevant documents of text corpus, instead, requires the most relevant 1000 documents returned.

According to these characteristics of Robust track, we adopt a two-step retrieval scheme. **The Key Notion** is as follow:

> ***Step 1*** *is a Boolean retrieval process to get a relevant document pool with "core query words" as input query. With the relevant document pool as the retrieval corpus,* ***Step 2*** *is a refining retrieval process with "core query words with expansion" as input query.*

In the scheme, the retrieval processing is divided into the following two steps as Figure 1.

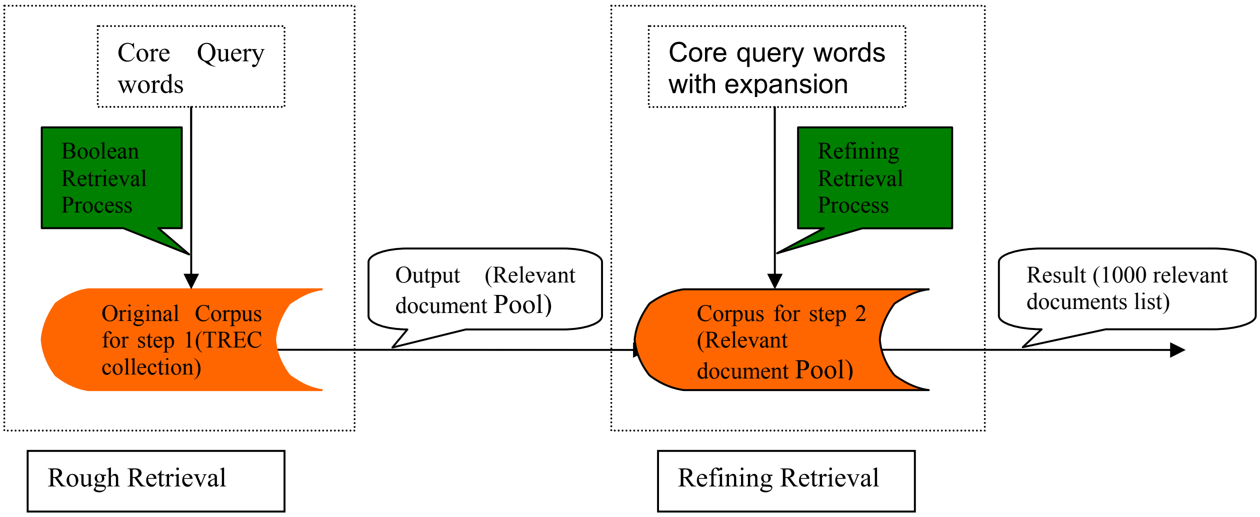

Core Query words
Boolean Retrieval Process
Original Corpus for step 1(TREC collection)
Output (Relevant document Pool)
Core query words with expansion
Refining Retrieval Process
Corpus for step 2 (Relevant document Pool)
Result (1000 relevant documents list)
Rough Retrieval
Refining Retrieval

Figure 1: Two-step Retrieval Scheme

Step 1, **Rough Retrieval**: A Boolean retrieval process is conducted based on the query words of the "title" field. This step is called as "rough retrieval". The returned relevant documents, which are called as "relevant document pool", will be the retrieval corpus of the second step.

Step 2. Refining Retrieval: In this step, we use the "relevant document pool" output from Step 1 as the retrieval corpus, and implement a series of methods to improve the ranking performance. This step is called as "refining retrieval". Specially, for this year's track, we try to use all fields of topic include "title", "description" and "narrative" as retrieval input in this step, and furthermore, we will tune the query weighting by pseudo feedback technique, windows-based model [Qianli Jin 2003] and other new techniques we presented this year.

The two-step retrieval scheme can reduce the influence of noise, because we get the probably relevant documents in step 1, which is the retrieval corpus for step 2. Then although there are a lot of noises for retrieval in step 2, it will not cause new irrelevant documents.

Another advantage is that retrieval cost is reduced, because the amount of documents for refining retrieval is much fewer than that of the whole text corpus.

2.2. Word Sense Entropy

Since selecting the most appropriate sense for an ambiguous word in a sentence is deemed to be of great benefit to Natural Language Processing, many researchers have tried applying Word Sense Disambiguation to information retrieval tasks. However, there are some problems for introducing WSD into IR. First of all, the precisions of the up-to-date WSD technologies are still not high enough. Furthermore, WSD-based IR will introduce extra system cost. Therefore, we attempted an alternative method to implement WSD idea.

As we know, an important part for various retrieval models is how to estimate the weight of query words, namely how to describe the importance for retrieval of different query words. Then we tried to introduce WSD idea into retrieval model by altering weight of query words. The common measure methods including TF, IDF, BM25 etc, are proved to be efficient in previous practical experiments. However, these methods are all empirical, and the weights of words are not independent on different corpus. The weight for one word might be different for different retrieval tasks, And in many instances, the weight of some word is not reasonable due to the incompletion of the corpus, which is also the common problem for Statistical NLP methods. For example, it is reasonable that the two words: ‘polio’ and ‘bank’, should have different weights because ‘polio’ is more distinctive for IR scoring than ‘bank’. But if the two words have both occasionally appeared only once in some corpus, then the IDF weights of the two words will be the same, 1.

Based on the above analysis, in the paper we proposed a new measure to weight the importance of query items—word sense entropy, which measures the variety of query word senses based on Wordnet’s structured knowledge. In the actual retrieval model, this weight is combined with other weight such as TF, IDF to weight the importance of query words.

As we know, Wordnet is a lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory [Wordnet 2.0]. Wordnet has provided detailed word sense information about English words. Some words have only 1 sense, and the others have several senses. Figure 1 is two examples in Wordnet.

"polio" has 1 sense:

> *Overview of noun polio*
> *The noun polio has 1 sense (first 1 from tagged texts)*
> *1. (1) poliomyelitis, polio, infantile paralysis, acute anterior poliomyelitis -- (an acute viral disease marked by inflammation of nerve cells of the brain stem and spinal cord)*

"bank" has 10 senses (first 9 from tagged texts):

> *Overview of noun bank*
> *The noun bank has 10 senses (first 9 from tagged texts)*
> *1. (883) depository financial institution, bank, banking concern, banking company -- (a financial institution that accepts deposits and channels the money into*
> *lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home")*
> *2. (99) bank -- (sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents")*
> *3. (76) bank -- (a supply or stock held in reserve for future use (especially in emergencies))*
> *4. (54) bank, bank building -- (a building in which commercial banking is transacted; "the bank is on the corner of Nassau and Witherspoon")*
> *5. (7) bank -- (an arrangement of similar objects in a row or in tiers; "he operated a bank of switches")*
> *6. (6) savings bank, coin bank, money box, bank -- (a container (usually with a*
> *slot in the top) for keeping money at home; "the coin bank was empty")*
> *7. (3) bank -- (a long ridge or pile; "a huge bank of earth")*
> *8. (1) bank -- (the funds held by a gambling house or the dealer in some gambling games; "he tried to break the bank at Monte Carlo")*
> *9. (1) bank, cant, camber -- (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)10. bank -- (a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning); "the plane went into a steep bank")*

Table 1: An Example from WordNet

It sounds reasonable that one word with fewer senses should be more important for retrieval than one word with more senses. According to this idea, we introduce word sense entropy to describe the word sense variety of one word. **The Key notion** is as follow:

> *One word with fewer senses should be more important for retrieval and have higher weight than one word with more senses.*

This weight could be used to the formula for scoring the rank of documents.

$$H(W) = \sum_{i=1}^{n} p(sense_i / W) \log(p(sense_i / W)) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$p(sense_i / W) = {c(sense_i, W)} \Big/ {c(W)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$H(W)$: Word sense entropy of word W ;

n: The amount of word senses in Wordnet of word W;

$sense_i$: The ith word sense in Wordnet of word W;

$c(sense_i, W)$: The frequency of w tagged as s_i in tagged texts;

$c(W)$: The total frequency of w in tagged texts.

In the actual retrieval model, this weight is combined with other weight such as TF, IDF to compute the similarity of two documents. For example, for simple TF*IDF retrieval model, the formula introducing word sense entropy is as the below:

$$R(q,d)=\sum_{Word_j\in(q\wedge d)} tf(Word_j)*idf(Word_j) \text{ .... original TF*IDF formula}$$

$$R(q,d)=\sum_{Word_j\in(q\wedge d)} tf(Word_j)*idf(Word_j)*H(Word_j)$$

.... Amended TF*IDF formula introducing word sense entropy

Where $R(q,d)$ denotes the similarity value between the query q and the document d.

2.3. Merging Multiple Runs

Since there are many different retrieval approaches, an old saying “two heads are better than one” could be used in our system to further improve performance. The ideal solution is to make use of multiple IR approaches and create a Meta IR engine whose core is a merging mechanism. In this year’s experiment, we have tried several different strategies merging multiple runs produced by different retrieval approaches.

The problem for merging multiple runs is described as follows: [Nick Craswell etc. 1999]

> *A document ranking $R=<D,o>$ consists of a set of documents D and an ordering o. Given N ranking $R_1\cdots R_N$, generate a single ranking $R_m=<D_m,o_m>$ such that $D_m=D_1\cup\cdots\cup D_N$ and o_m is an effective ranking, meaning that it tends to rank relevant documents above irrelevant ones.*

2.3.1.Merging Simply Several Runs

Suppose that we have 10 run results. First, we select the documents which appear in each of the returned document lists of 10 runs, to become the firstling members of the merging result; then, choose the documents which appear in 9 runs; finally we choose the documents in turn from each of 10 runs which rank is the highest of the remained documents in the specific run. Repeat the final choosing process until the amount of documents of merging result is 1000,

2.3.2.Merging by Score Normalization

The first merging method is simple but proves useful in improving the precision of retrieval as our experiences; however it requires enough runs produced by different retrieval methods, and the system cost is high. Therefore, we continue to consider how to efficiently merge small amount of runs such as only two runs. The direct idea is to normalize the different scores produced by different retrieval methods to the scores able to be compared with each other. We refer to the normalization methods in statistical theory and experiment Max-Min normalization and Normal normalization. (Although the following experiences aim at merging the results of only two runs, the merging schemes could also be practical for merging the results of beyond 2 runs)

For Max-Min normalization formula,

$$score'_i = \frac{score_i - score_N}{score_1 - score_N} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

$score'_i$: Normalized score for the ith document;

$score_i$: Score of the ith document in this run;

$score_1$: Score of the 1st document in this run;

$score_N$: Score of the last document (Nth) in this run. N = 1000 in Robust track

For Normal normalization formula,

$$score'_i = \frac{score_i - \overline{score}}{\sigma} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

$score'_i$: Normalized score of the ith document;

$\overline{score}$: Mean score of the 1000 relevant documents in this run;

$score_i$: Score of the ith document in this run;

σ : Variance of the 1000 relevant documents in this run.

After normalizing the scores of different runs according to formula (3) and (4), we can select the top 1000 documents from these two runs according to the ranking sequence of normalized scores.

2.3.3.Merging by Clustering

Because each retrieval methods' scoring methods is different, the intuitive explanation is not clear for normalizing retrieval scores. We experiment another merging method, which is based on result documents clustering.

As we know, the documents could be relevant with each other with great probability, if they are relevant with the specific topic. So if some documents retrieved by different retrieval methods are

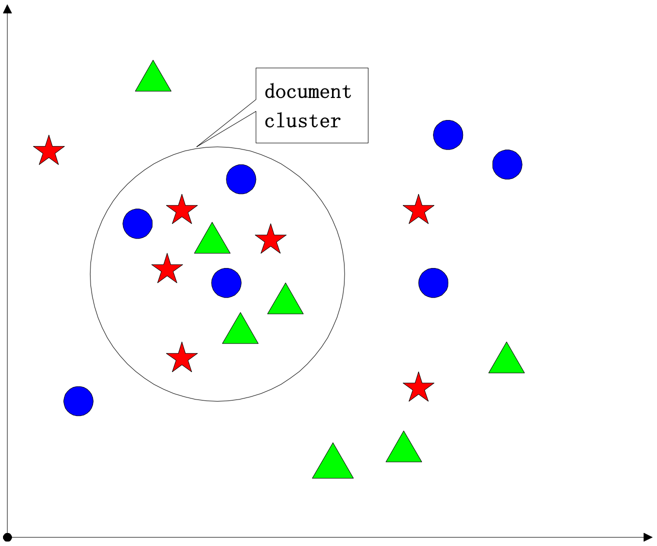


Figure 2: Merging by clustering

Where Triangle, Star, Circle present the documents from different runs.

similar with each other, then we believe that these documents are more relevant with this topic, ie. the probability, that these documents is relevant with this topic is higher than that of other documents in those runs. **The Key notion** is demonstrated as right figure (figure 2):

According to this idea, we design a merging algorithm (in our experiment the amount of runs is 2):

Assuming the merged document set is $D = \{d_i\}, i = 1, \cdots N$, $1000 \le N \le 2000$

a) Select the documents which appear in both runs as a part of the new merging result. $D_1 = \{d_{1k_1}\}$, $k_1 = 1, \cdots, M_1$

b) D_1 is used as the document pool which is benchmark for comparing the similarity of other documents.

c) Compute the similarity scores of the remained documents $D - D_1$ and D_1. Select the documents of top similarity scores as the new merging result $D_i = \{d_{ik_i}\}$, $k_i = 1, \cdots, M$, M is usually 2.

d) Define the document set D_i in step c as the new document pool of benchmark.. Repeat the process of step c: Compute the similarity scores of the remained documents $D - \bigcup_{j=1}^{i} D_j$ and D_i. Select the documents of top similarity scores as the new merging result $D_{i+1} = \{d_{i+1k_{i+1}}\}, k_{i+1} = 1, \cdots, M$

e) Repeat step c and d until the amount of documents is 1000.

3. Experimental Results

In this year's track, we create a baseline run "NLPR04OKapi" based on BM25 retrieval method, a baseline run "NLPR04SemLM" which uses windows-based model on word and sense entropy weighting and a baseline run "NLPR04LMts" which uses windows-based model, feedback technique in Lemur toolkit and two-step retrieval scheme. These runs all use three fields of topics: "title", "description" and "narrative".

Furthermore, for the experiment of merging methods, we use the lemur toolkit which is for language model and information retrieval to create 4 baseline runs [6][7]: KL-DIR, KL-DIR-DIVMIN, TWO-STAGE and JM smoothing. (Detailed information about lemur can be seen in http://www-2.cs.cmu.edu/~lemur/).

3.1. Experimental Data

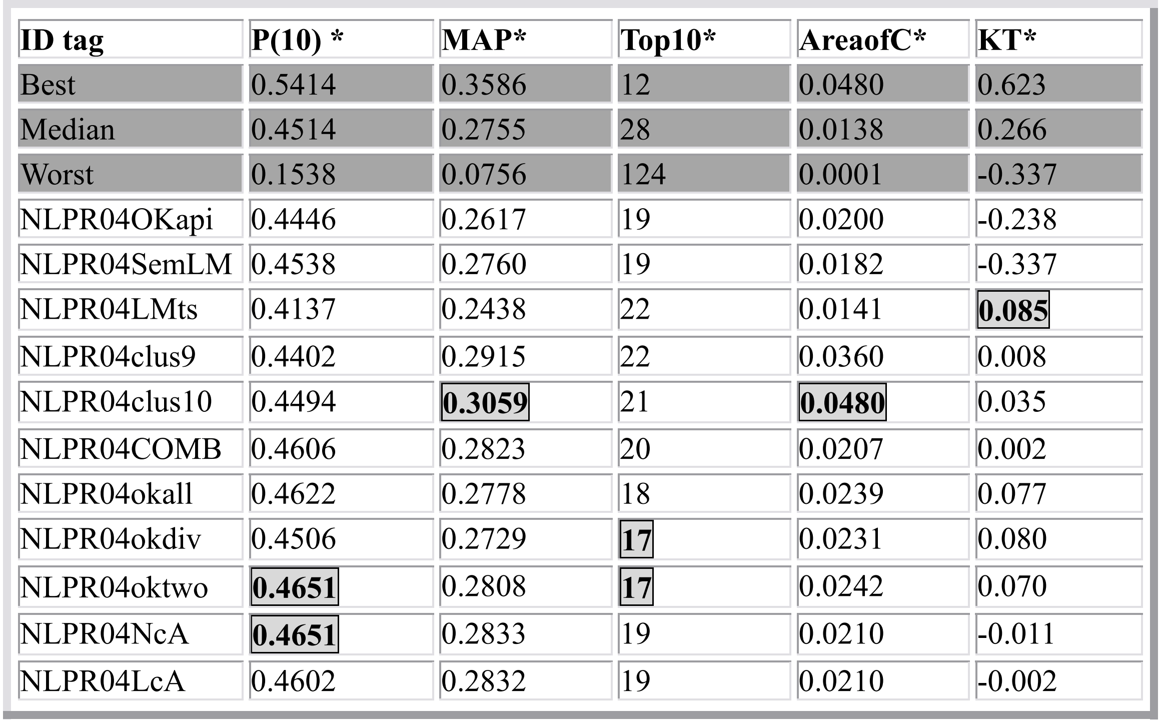
The followed table 1 is the description for our submitted runs in this year's robust track.

ID tag	**Description**
NLPR04OKapi	Baseline retrieval system using BM25
NLPR04SemLM	Baseline retrieval system using windows-based

	model and Word Sense Entropy weighting
NLPR04LMts	Baseline retrieval system using windows-based model, feedback technique in Lemur toolkit and two-step retrieval scheme
NLPR04clus9	Simply merge the 9 runs created by different methods
NLPR04clus10	Simply merge the 10 runs created by different methods
NLPR04COMB	Merging by Clustering with NLPR04SemLM and KL-DIR
NLPR04okdiv	Merging Okapi and KL-DIR-DIVMIN in Max-Min normalization
NLPR04okall	Merging Okapi and KL-DIR in Max-Min normalization
NLPR04oktwo	Merging Okapi and TWO-STAGE in Max-Min normalization
NLPR04NcA	Merging Okapi and KL-DIR-DIVMIN in Normal normalization
NLPR04LcA	Merging SemLM and KL-DIR in Normal normalization

Table 2: Description for our submitted runs

The followed table 2 is the evaluation comparison over all topics for our submitted runs in this year's robust track.



ID tag	P(10) *	MAP*	Top10*	AreaofC*	KT*
Best	0.5414	0.3586	12	0.0480	0.623
Median	0.4514	0.2755	28	0.0138	0.266
Worst	0.1538	0.0756	124	0.0001	-0.337
NLPR04OKapi	0.4446	0.2617	19	0.0200	-0.238
NLPR04SemLM	0.4538	0.2760	19	0.0182	-0.337
NLPR04LMts	0.4137	0.2438	22	0.0141	**0.085**
NLPR04clus9	0.4402	0.2915	22	0.0360	0.008
NLPR04clus10	0.4494	**0.3059**	21	**0.0480**	0.035
NLPR04COMB	0.4606	0.2823	20	0.0207	0.002
NLPR04okall	0.4622	0.2778	18	0.0239	0.077
NLPR04okdiv	0.4506	0.2729	**17**	0.0231	0.080
NLPR04oktwo	**0.4651**	0.2808	**17**	0.0242	0.070
NLPR04NcA	**0.4651**	0.2833	19	0.0210	-0.011
NLPR04LcA	0.4602	0.2832	19	0.0210	-0.002

Table 3: Evaluation comparison over all topics for our submitted runs

3.2. Experimental Analysis

According to the above experimental results, we can get the following conclusions.

I) Compared with our last year's submission [Qianli Jin 2003], our baseline results ("NLPR04SemLM" and "NLPR04LMts") of this year perform favorably. It shows that the Two-Step scheme and Word Sense Entropy Weighting techniques are efficient.

II) Compared with our baseline systems of this year, the merging methods prove efficient. The simple merging method's run ("NLPR04clus10") gets the highest MAP.

III) For merging methods by score normalization, they improve the performance of baseline systems; however we could find that the nuance between max-min ("NLPR04okdiv", "NLPR04okall", "NLPR04oktwo") and normal normalization ("NLPR04NcA", "NLPR04LcA") methods' improvements over the baseline system is slight. The reason might be the two score normalization methods are both not clear in intuitive explanation and the two normalization methods have not obvious predominance to each other.

IV) The run of merging method by clustering ("NLPR04COMB") also perform favorably in comparison with the baseline system of this year, but because we submit only one run, the experiment maybe not sufficient enough to prove the efficiency of the clustering method.

V) For predicting the hardness of topics, it is a fiasco for our runs. We have misunderstood this sub track. The track has required a strict ordering of all 250 topics in the test set from easiest (1) to most difficult (250). However we have just reversed the ordering: from most difficult (1) to easiest (250). That is why we have got the worst score in Predicting the hardness of topics. Our prediction for hardness is simple, which only use the distribution of score for each run. This process for hardness prediction is proved somewhat simple.

4. Conclusion

Three techniques are introduced in this year's robust track: (1) two-step retrieval scheme; (2) word sense entropy; (3) several strategies for merging multiple runs. And the experiments prove that these techniques are efficient to some extent in improving the performance of worst-query. Unfortunately, for predicting the topic hardness, we misunderstand it and get a poor performance.

5. Acknowledge

This work was supported by the Natural Sciences Foundation of China under grant No. 60372016, the Natural Science Foundation of Beijing under grant No. 4052027 and the Scientific Research Foundation for Returned Overseas Chinese Scholars, State Education Ministry.

Our work have used two web-shared toolkits: Lemur toolkit for language model and information retrieval (URL: http://www-2.cs.cmu.edu/~lemur/), and Wordnet 1.7 (URL: http://www.cogsci.princeton.edu/~wn/).

6. Reference

[1] Qianli Jin, Jun Zhao, Bo Xu. *NLPR at TREC 2003 – Novelty and Robust Track*. Text Retrieval Conference (TREC-12), NIST, Maryland, USA, 2003.

[2] Nick Craswell, David Hawking, Paul Thistlewaite. Merging Results From Isolated Search Engines. Australasian Database Conference, 1999

[3] Qianli Jin, Jun Zhao, Bo Xu, *Window-based Method for Information Retrieval*, 2004, The First International Joint Conference on Natural Language Processing

[4] Ellen M. Voorhees, *Overview of the TREC 2003 Robust Retrieval Track.* Text Retrieval Conference (TREC-12), NIST, Maryland, USA, 2003.

[5] http://www.cogsci.princeton.edu/~wn/

[6] Chengxiang Zhai and John Lafferty. *Model-based feedback in kl divergence retrieval model*. In Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM), 2001.

[7] John Lafferty, Chengxiang Zhai, *Risk minimization and language modeling in information retrieval.* In Proc. Of 24th ACM SIGIR, 2001