# WIM at TREC 2005*

Junyu Niu, Lin Sun, Luqun Lou, Fang Deng, Chen Lin, Haiqing Zheng, Xuanjing Huang
Lab of Web Information Mining, Computer Science & Engineering Department, Fudan University
Shanghai 200433, China
{jyniu,sunl,032021211,032021202,042021182,042021175,xjhuang}@fudan.edu.cn

**Abstract**
This paper describes the three TREC tasks we participated in this year, which are, Genomics track's categorization task and ad hoc task, and Enterprise track's known item search task. For the categorization task, we adopt a domain-specific terms extraction method and an ontology-based method for feature selection. A SVM classifier and a Rocchio based two staged classifier were also used in this experiment. For the ad-hoc task, we used BM25 algorithm, probabilistic model and query expansion. For the Enterprise track, language model was adopted, and entity recognition was also implemented in our experiment.

**Keywords:** Information retrieval, text categorization, domain-specific terms extraction, ontology, SVM, probabilistic model, entity recognition, Rocchio

## 1. Introduction

WIM participated in Genomics track and Enterprise track in TREC 2005. This year's Genomics track consists of two tasks, one is called ad-hoc retrieval task, the other is called categorization task. In the ad hoc task of Genomics track, we mainly concern on: (1) the efficiency of language model; (2) query expansion (3) the weight of query terms. In this year's Enterprise Track, we attended Known Item Search task.

This paper is organized as follows. First, we give a detailed description of our experiment on categorization task. We present the main architecture of our system and discuss every step independently and carefully. Then we discuss our work on ad hoc task. Finally, we describe our system for enterprise track of this year.

## 2 Categorization task of Genomics Track

The categorization task is similar to last year's triage task [1] in the purpose of classifying the articles collected from MGI correctly for the curators for exhaustive analyses. Those articles should be classified to four categories: Tumor biology, Embryologic gene expression, Alleles of mutant phenotypes and Gene Ontology. So we can regard this task as a multi-class classification task. A lot of approaches have been discussed and published on machine learning, text filtering and so on [2] , those techniques were carefully examined, and some of those which adopted in our research will be represent in this paper later.

The data set of this task is collected from three magazines in the biochemistry field which contains 11880 documents. And those articles are in SGML format. The task is to find out the articles to be sent to the curators for manual operation. Those articles are regarded as the "positive" samples, while the others are treated as the "negative" samples. And the official measurement for this categorization task is the utility score which just like last year.

We have submitted twelve runs of this task which will be discussed carefully later, and got highest scores of E and G subtasks on the feature generated by the feature selection method based on domain-specific term extraction using corpus comparison.

### 2.1 System description

As figure 1 shows, the system contains three main parts: cleaning, feature selection, and classification.

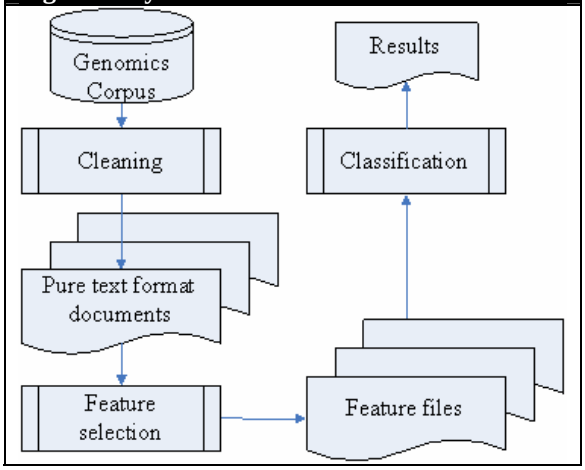
In the cleaning part, a SGML parser was developed to transform the corpus files into pure text format files. The document's body text, keywords, and glossary were extracted. We experimented using different parts of the document for feature selection and classifying. Terms were separated by the punctuation and blank characters, the hyphen characters and the full stop characters between figures or characters were ignored too. Porter stemmer and different stopword lists were implemented.

*Supported by the National Natural Science Foundation of China (No. 60305006)

In the feature selection part, two feature selection methods were implemented, one of which focused on domain-specific term extraction using corpus comparison, the other focused on word-meaning and the usage of domain-specific ontology.

In the classification part, two-stage classification strategy was used. Classifiers such as $SVM^{Light}$ [3], NN, KNN, and Rocchio were implemented on different features to find out the best combination of classifier and feature.

**Figure 1: System architecture**




## 2.2 Feature selection based on domain-specific term extraction using corpus comparison

As known, terms of genomics domain are generally more important for representing the documents of genomics corpus. The traditional methods which rely on genomics domain knowledge databases or dictionaries could not reflect the corpus' features precisely. These methods have three main limits. (1) It's a labor-intended job to create such knowledge databases or dictionaries. (2) Processing full length documents of large corpus according to these databases or dictionaries is time consuming. (3) These methods heavily rely on domain-specific knowledge databases or dictionaries and can't be applied easily to any new domains.

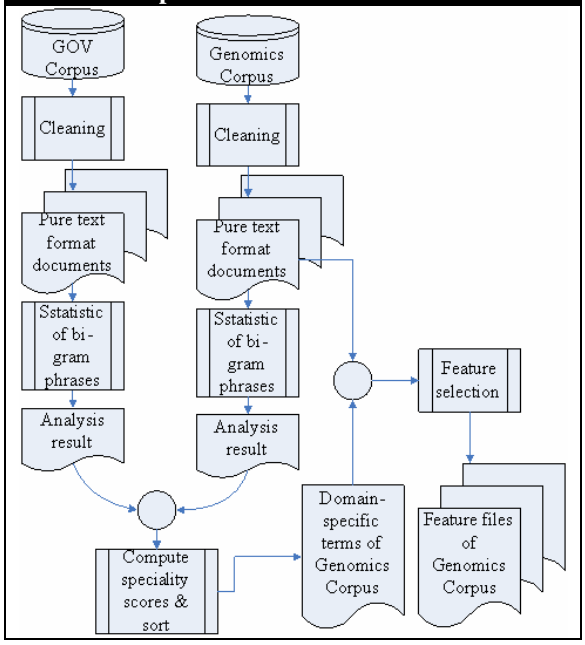
To extract domain-specific terms without these limits, we assume that the distribution of terms following the domain-specific terms varies in different domain corpus, and the larger variation indicates the larger speciality of the domain-specific terms.

Based on this assumption, we selected the GOV corpus as the general domain corpus, for each term, we compared the distribution of terms following it then the speciality score is computed out, terms with top speciality were selected as domain-specific terms. Finally, documents were represented by these domain-specific terms and terms around them.

Our experiment has shown good results. And the performance is acceptable. The genomics domain-specific terms in genomics corpus can be computed out in some minutes.

The architecture of this method is shown in the following figure 2.

**Figure 2: Architecture of feature selection based on domain-specific terms extraction**




### 2.2.1 Corpus selection

To select a corpus for comparing, there are two basic requirements in corpus selection: (1) the selected corpus should be in general domain, contains wide range topics; (2) the selected corpus should contain similar number of terms as the genomics corpus.

However, it's hard to follow the second requirement strictly. We selected a subset of the GOV corpus with similar size instead of similar number of terms as the genomics corpus, which can meet the second requirement approximately.

### 2.2.2 Bi-gram phrase analysis and computing the speciality scores using corpus comparison

Bi-gram phrase is defined as two terms without any terms or punctuation between them.

In order to compute the speciality scores of each term, the same terms were assigned a unique ID in two corpuses. Then, the following statistics were counted:
(1) $tf_i^A$: frequency of term i in Genomics Corpus.
(2) $tf_i^B$: frequency of term i in GOV Corpus.
(3) $pf_{ij}^A$: frequency of bi-gram phrase begin with term i and end with term j in Genomics Corpus.
(4) $pf_{ij}^B$: frequency of bi-gram phrase begin with term i and end with term j in GOV Corpus.
(5) $fn_i$: the number of different terms following term i in bi-gram phrases in both Corpuses.

The speciality scores of terms are calculated by the following formula:

$$S_i^A = \sum_{j=1}^{fn_i} \left(\frac{pf_{ij}^A}{tf_i^B} - \frac{1}{fn_i}\right)^2$$

$$S_i^B = \sum_{j=1}^{fn_i} \left(\frac{pf_{ij}^B}{tf_i^A} - \frac{1}{fn_i}\right)^2$$

$$score_i = \frac{S_i^A}{S_i^B}$$

$score_i$: speciality score of term i.
$S_i^A$, $S_i^B$: temporary variables
Other symbols have been illustrated above.

According to this formula, terms with $fn_i=0$ are ignored, terms with $S_i^B=0$ *and* $S_i^A\neq 0$ are assigned -1 to indicate maximum value. Terms with higher speciality score are selected as genomics domain-specific terms.

### 2.2.3 Feature selection with domain-specific terms

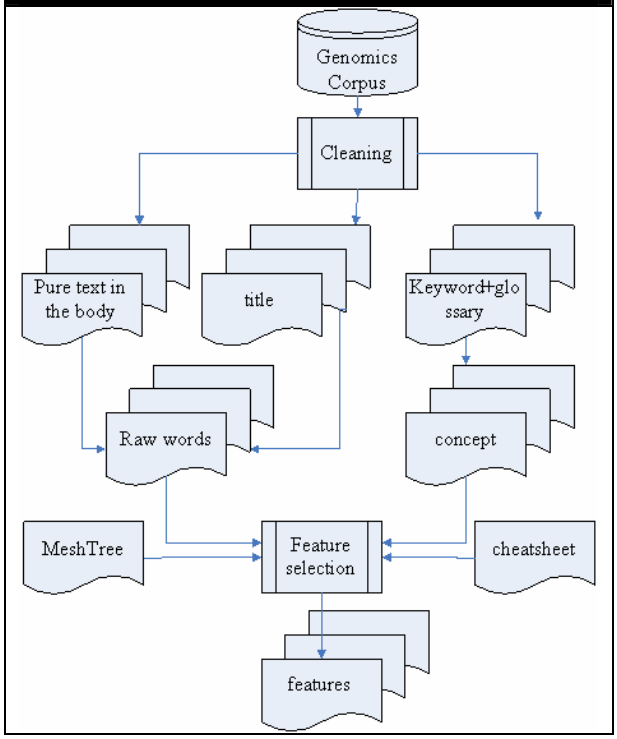
The selected genomics domain-specific terms and 2 or 4 terms before or following them are selected to represent the document. In order to see how well this method works, we simply use single terms as features. We compared the raw term frequency and log term frequency as the feature value, and found out that log term frequency worked much better than the raw term frequency.

## 2.3 Ontology-guided Feature Selection

For large corpus multi-class classification, it is a challenge to select a certain amount of informative features to represent the documents. High dimension of features will distinctly reduce the performance of classifiers. General text categorization methods, which treat document as "bag of words" and compare the term-goodness by statistic information rather than semantic information, lead to poor understanding of documents. Furthermore, recognizing the entities is a key problem for biomedical text categorization. The meanings of the entities and the inherent hierarchical structure of the nomenclature are not appropriately treated in those approaches. Our research shows that feature selection employing a domain-specific ontology has a promising effect in solving these problems.

Figure 3 describe the architecture of this ontology guided feature selection.

**Figure 3: Architecture of the ontology guided feature selection**




The paragraphs in the body text and captions of the figures in biomedical essays are regarded as informative. Therefore words from these parts were selected. Keyword and glossary are special parts, while both of them are helpful to define the key concepts discussed in the document. Hence keywords and glossaries were selected as features

too. For dimension reduction, we abandoned the frequently used methods, such as document frequency, chi-square and information gain, which all define a threshold to select an acceptable number of features for classifiers. An ontology-guided approach was adopted instead. We made advantage of the medical ontology MESH_Tree and manmade rules concluded from the cheatsheet to reduce the influence of synonyms and hyponyms. Synonymous terms were removed and entities were changed to a more general form according to MESH_Tree. After the two steps, each feature was given a weight related to its distribution in the corpus. The weight of term i in document k was computed by a formula slightly different from the original entropy formula.

$$a_{ik} = \log(f_{ik} + 1.0) * (1 + \frac{1}{\log N} \sum_{j=1}^{N} [\frac{f_{ij}}{\frac{n_i}{df_i}} \log(\frac{f_{ij}}{\frac{n_i}{df_i}})])$$

**2.4 Classifiers**

Classifiers we used include the SVM$^{Light}$, Neural Network, KNN, and Rocchio. Our experiments show that the SVM$^{Light}$ classifier is suitable for the feature generated by the domain-specific term extraction method, while the Rocchio classifier is suitable for ontology based method.

After having researched the corpus carefully we realized that the four classes of this year's Genomics track's categorization task have there specialty. Taking the T(tumor) class for example, there are some specialties from the aspect of word-building. Some words always indicate that those articles having strong relationship with the class T. While some others indicate that this article must have no or less relationship with that class. All the documents in the corpus are filtered by those rules. After that, we got a small corpus.

| Corpus Size | Before Filtering | After Filtering |
|---|---|---|
| | 6043 | 2494 |

Table 1: The number of the test files before and after filtering

For the second stage classifier in the year's categorization track, we used a traditional Rocchio algorithm. The Rocchio algorithm was used a lot in the relevance feedback in information retrieval area [4]. This algorithm first formed a center of each class, and then computed the similarity between all the samples in the test set and that center vector. Then we did classification according to the similarity value. We will describe the Rocchio classifier implemented in our experiment in the following part.

After the feature selection step a sample in the corpus was represented by a vector such as $F = (< t_i, w_i >, ..., < t_m, w_m >)$ , in which $t_i$ stands for the term of indexed files, and $w_i$ is the corresponding weight to the term $t_i$. Then we form an initial profile for these training samples. So the center of each class is as follows: $Center_i = (w_{i1}, w_{i2}, \cdots, w_{im})$, where the weight of the feature term $t_{ij}$ is

$$w_{ij} = \alpha \sum_{d_l \in pos_i} w_{il} - \beta \sum_{d_l \in neg_i} w_{il}$$

in which $pos_i$ means the set of positive training samples for class i , and $neg_i$ means the negative training samples for the class i. $\alpha$ , $\beta$ are real numbers, which indicate the importance of positive and negative portion of the training samples when forming the center of each class.
When the center for each class was formed, we can compute the similarity between the documents to be classified and each class's center vector.

$$Sim(d_t, Center_i) = \frac{\sum_{l=1}^{m} w_{tl} w_{il}}{\sqrt{\sum_{l=1}^{m} w_{tl}^2} \sqrt{\sum_{l=1}^{m} w_{il}^2}}$$

At last, a threshold for each class was defined, and those documents with similarity value bigger than the threshold was classified as the positive ones, while those with smaller similarity values were be treated as the negative ones.

And we also do some slightly change to the equation we mentioned before, which means that we did not use all the negative samples. We notice that those documents which are more similar to the positive samples have stronger effect to the classification results. So, we formed a new set called Nearly Positive Set, presented as $npos_i$ . In our experiment we treated the articles which

derived from the filtering step which applied those decision rules we described as our first stage classifier. So we get the new equation to form the center of each class as follows:

$$w_{ij} = \alpha \sum_{d_l \in pos_i} w_{il} - \beta \sum_{d_l \in npos_i} w_{il} .$$

Table 2 shows the final utility of our experiment, which shows that about 3% increment on normalized utility when we use the near positive samples.

Here we want to point out that in our categorization task, the negative samples are really important. We have done some experiments which showed that if we just ignore the negative samples, the performance will be really bad when compared to the runs which when using the negative samples which are in the near positive set. From table 2 we will see that the normalized utility of the latter method improves nearly about 15%.

| | A | E | G | T |
|---|---|---|---|---|
| Rocchio( without negative samples) | 0.6440 | 0.6519 | 0.5595 | 0.7444 |
| Rocchio( with all the negative samples) | 0.8095 | 0.8113 | 0.5691 | 0.8550 |
| Rocchio( with NPOS set samples) | 0.8168 | 0.8207 | 0.5765 | 0.8677 |

Table 2: The utility score of each class for different classifiers (these data derived from thresholds which adjusted from those used for official runs)

In this year's track, we have combined the two feature selection methods with different classifiers, such as Neural Network, k-nearest neighbor, SVM$^{Light}$, Rocchio and so on, from which we found that when combined with the domain-specific term extraction the SVM$^{Light}$ classifier produced the best results. The parameters for this classifier we adopted were just the same as Fujita [5].

**2.5 Results**

We submitted 3 series of runs for each subtask, totally 12 official runs. Details are listed in the following table 3.

| | A | | |
|---|---|---|---|
| | P | R | NU |
| MarsI | 0.4754 | 0.9006 | 0.8421 |
| MarsII | 0.4195 | 0.9187 | 0.8439 |
| MarsIII | 0.3254 | 0.9096 | 0.7987 |
| | E | | |
| | P | R | NU |
| MarsI | 0.1899 | 0.9333 | 0.8711 |
| MarsII | 0.1899 | 0.9333 | 0.8711 |
| MarsIII | 0.0794 | 0.9524 | 0.7799 |
| | G | | |
| | P | R | NU |
| MarsI | 0.2644 | 0.778 | 0.5813 |
| MarsII | 0.2122 | 0.8861 | 0.587 |
| MarsIII | 0.191 | 0.9093 | 0.5591 |
| | T | | |
| | P | R | NU |
| MarsI | 0.1061 | 0.95 | 0.9154 |
| MarsII | 0.099 | 0.95 | 0.9126 |
| MarsIII | 0.0286 | 1 | 0.8528 |

Table 3: The results of official runs.

MarsI: we select the best runs for each subtask in our experiments, their classifier are all SVM$^{Light}$, the features are all generated by the feature selection method based on domain-specific term extraction using corpus comparison.

A. window size is 4, top 2000 specified domain-specific terms, except terms with the speciality score of -1.
E. window size is 2, top 500 specified domain-specific terms, except terms with the speciality score of -1.
G. window size is 4, top 2000 specified domain-specific terms, except terms with the speciality score of -1.
T. window size is 0, top 25000 specified domain-specific terms, including terms with the speciality score of -1, about 5000 terms with lowest otherness score added to the stopword list.

MarsII: features are generated by the feature selection method based on domain-specific term extraction using corpus comparison. We fix the number of domain-specified terms as 500, except terms with the speciality score of -1, and the size of window as 2, use SVM$^{Light}$ classifier, to see how it works for different subtask. This series of runs comes out the highest mean normal utility score of our three series of runs.

MarsIII: we employed a domain-specific ontology

for feature selection and compute the entropy for each term selected from our last step. And at last we implemented a two-stage classifier for this categorization job.

"Window" is defined as the selected domain-specific terms together with terms around them.

"Window Size" is defined as the number of terms before or following the selected domain-specific term.

Table 4 gives a glance of our best results among all official results of this year's categorization task.

| | Best | Median | Worst | Our Best | Runs |
|---|---|---|---|---|---|
| A | 0.8710 | 0.7785 | 0.2009 | 0.8439 | 48 |
| E | **0.8711** | 0.6548 | -0.0074 | **0.8711** | 46 |
| G | **0.5870** | 0.4575 | -0.0342 | **0.5870** | 47 |
| T | 0.9433 | 0.7610 | 0.0413 | 0.9154 | 51 |

Table 4: Results

### 2.6 Conclusions and future work of Categorization Task

The results of our official runs have shown that the methods mentioned above worked well.

For the domain-specific terms extraction method, the results have shown that the assumption is reasonable, the extracted terms can represent the domain-specific documents very well, and are fitter for the Embryologic Gene Expression class among the four classes.

The ontology-guided feature selection method also has positive affect. It has reduced the dimension and improved the system performance successfully.

And in our experiment we found that some classification methods are really sensitive to the feature which selected from the initial corpus and the dimension of the feature space. After compared a lot classifiers, such as SVM, Neural Network and so on, we found that the performance of the Rocchio classifier seems really stable.

We believe that the combination of these two feature selection methods will generate better features and lead to better results. This is what we plan to do in the future work.

## 3. Ad Hoc retrieval task of Genomics Track

This year we also attended genomics ad hoc, of which the queries changed a bit, making it much closer to normal requirement than before.

The ad hoc retrieval task was designed to simulate the subject topic retrieval against a ten year subset (4,591,008 records) of the MEDLINE bibliographic database as 2004. This year, there were fifty official (and other samples) search topics derived from interviews on real biology researchers, taking the form of QA (question && answer) rather than phrases.

Relevance assessments were carried out by using the conventional pooling method, and all the pooled documents were divided into three genres: definitely relevant (DR), possibly relevant (PR) or not relevant (NR) against the information needs. Documents in the first two genres were considered relevant in official evaluations.

Participants were required to submit two sets of top 1000 relevance ranked lists of documents retrieved by either automatically or manually constructed queries from given search topics. There were no specific restrictions concerning using data resources, and we only chose some dictionaries as our main resources.

This year we mainly concerned on the difference between BM25 algorithm [6] and the KL-divergence algorithm [7] and give each dictionary a distinct weight.

### 3.1 System Description

We mainly used two sets of systems, okapi and probabilistic language model, which were quite popular in the past few years.

We extracted TI, MH,AD and AB from the Corpus, weighing respectively 1.0, 0.5, 0.5 and 1.0, and remarkably influencing the final result according to results of the previous years.

In the course, we also utilized the porter stemming and removed the stop words, and found that stemming could improve the MAP.

Furthermore, we also applied the blind feedback technology for Okapi [8] BM25 algorithm and KL-divergence for Language models.

BM25 TF was incorporated in the dot-product

matching function between TF*IDF weighted vectors. Typical parameters like k1, b could be adjusted.

It was first put forward and implemented by City University. And it proved that this algorithm could do very well on the WEB track.

This year, two values of the parameters in our system -- k1and b of our system were respectively 0.1 and 0.8.

Uses of probabilistic language model in information retrieval intended to adopt a theoretically motivated retrieval model. Ponte and Croft first applied a document unigram model to compute the probability of the given query generated from a document [9]. Furthermore, the probabilistic language model was also used by the FUJITA who achieved the best MAP score.

This year, the three parameters of out system--- the value of feedback coefficient, the number of terms feedback and the value of documents feedback were respectively 0.1, 10 and 100

### 3.2 Query Expansion

For each search topic, we held it was better for us to look up the categories of the terms if a question contained more than two genomics terms. If these terms belonged to one category, the weight should be added. For example, for two query terms in the human category, the weight of them should receive a bonus of 0.5.

| **Term Category** | **Bonus** |
|---|---|
| Others | 0.0 |
| Mouse | 0.2 |
| Human | 0.5 |

Table 5: Term Category and relevant bonus

We also weighed the dictionaries. Every dictionary should have a distinctive weight. Thesaurus looked up in one dictionary should bring the dictionary's weight, so even a query term that was extended, should have some different weight of thesauruses. But the result wasn't satisfying, probably due to the problems in the programming or the wrong method. This needs our efforts to improve.

### 3.3 Results

We submitted 50 search topics results, all of which got judged except the topic 135.

| Runs | Method | MAP | P10 | R-prec |
|---|---|---|---|---|
| Wim1 | BM25 | 0.1781 | 0.3347 | 0.2094 |
| Wim2 | KL- divergence | 0.1807 | 0.3000 | 0.2006 |

Table 6: the Results

Through this experiment, we conclude that the performance of KL-divergence is a bit better than that of BM25 algorithm.

### 3.4 Conclusions and future work of Ad hoc task

For the ad hoc retrieval task, we submitted one run using BM25 algorithm and another run using KL-divergence with Dirichlet smoothing. It seems that the run using KL-divergence with Dirichlet smoothing is better than the one using BM25 algorithm.

In future, we will continue the research on the query expanding methods and how to give a reasonable weight. Furthermore, we may improve the smoothing method used in the KL-divergence.

## 4. Enterprise Track

This year is the first year of this track. And there are few mature methods or model. We mainly focus on Known Item Search subtask of the email search. The scenario of this subtask is that the user is trying to find an important email that they know exists.

Among the 25 questions for training, we found there are 4 questions contained people names while 3 questions contained time information. For these 7 (28%) special questions, we tried to do entity recognition, which seems also effective for the test questions from our results.

### 4.1 Experiments

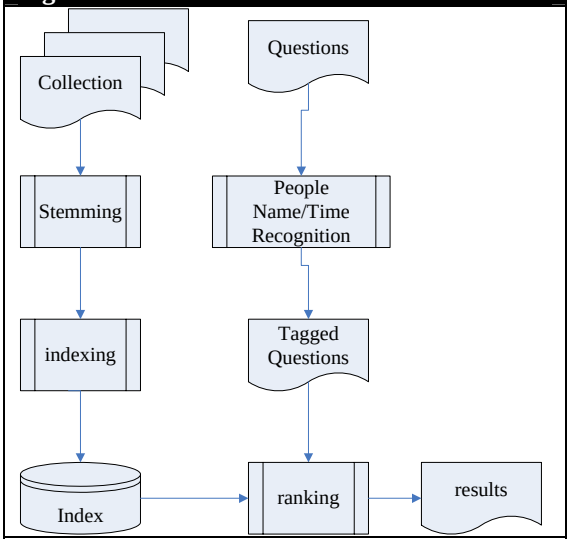
Figure 5 shows our system architecture. And the retrieval model is a unigram language-modeling algorithm based on Kullback-Leibler divergence [10].

We recognize people's name and time based on a set of rules. Both of the two kinds of information are looked as phrases. It means, when we do retrieval for these phrases, the nearer the words in phrase appear, the higher the score of the mail is.

We found that there are many questions related to people's name and time (table 7). There are 16

questions containing people name, and 3 questions containing time info.

**Figure 4: Architecture of KI**




| | Question numbers contain the info | average RR of that kind of questions (all runs) | average RR of that kind of questions (my run) |
|---|---|---|---|
| Time | 3 | 0.40 | 0.56 |
| people name | 16 | 0.767 | 0.862 |

Table 7: number of questions contained time/people name.

### 4.2 Results & discussions

The Average reciprocal rank of our submitted run is: 0.533. The figure 5 below is the ranking distributing figure of our system. 39% questions gain rank1. Number of topics for which target page found in top 10: 98 (78.4%)

The data related questions gain 40% better, and people's name contained questions gain 12.3% effects. So, people's name and time recognition can improve retrieval results.

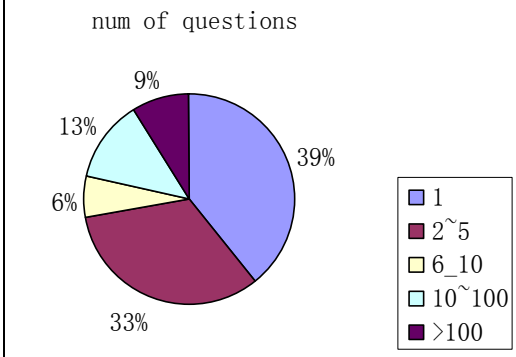
However, we could not do well for some questions. 6 questions are like QA question. For example, KI64: Why doesn't Amaya have a hand-shaped cursor? And the average RR of 67 submitted runs for the 6 questions are 0.059, 0.091, 0.125, 0.333, 0.5, and 1. So, normal method could not do well for this kind of questions. I think, in future we can make improves using two measures below:

A. More semantic understanding is needed.

Try to use some common methods in QA

B. The relation between original message and the reply messages.

**Figure 5: Question distributing figure**




Basically, there are two kinds of results people trying to get from the known-item search: one kind is announcements, which is always locate in the original message; the other kind is the answer of some questions, which always locate in the reply messages. For the second kind, the reply messages need more attention.

### References


[1] William R. Hersh. TREC 2004 Genomics Track Overview. Proceedings of the 13th Text Retrieval Conference, 2004

[2] Sebastiani, F., 2002, Machine Learning in Automated Text Categorization. ACM Computing Surveys (CSUR) Volume: 34 Mar, Issue: 1, 1-47

[3] Joachims, T. Making Large-Scale SVM Learning Practical. Advances in Kernel Methods – Support Vector Learning, B. Scholkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999

[4] Allan, J., 1996. Incremental Relevance Feedback for Information Filtering. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, Zurich, 270-280

[5] Sumio FUJITA, Revisiting Again Document Length Hypotheses TREC-2004 Genomics Track Experiments at Patolis, Proceedings of the 13th Text Retrieval Conference, 2004

[6] Robertson, S.E., Walker S., Jones S., Hancock-Beaulieu, M.M. and Gatford, M. 1995. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference(TREC-3), NIST Special Publication 500-225, Washington D.C., 109-126.

[7] Zhai, C. and Lafferty, J. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In Proceedings of the 2001 ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, 334-342.

[8] Robertson, S.E. and Walker S. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In Proceedings of the 1994 ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 232-241.

[9] Ponte, J. and Croft, W. B. 1998. A language modeling approach to information retrieval, In Proceedings of the 1998 ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 275–281.

[10] T.M. Cover and J.A. Thomas. Elements of Information Theory. New York: Wiley, 1991.