Experiments in TREC-2003 Genomics Track at NTT

Hirotoshi Taira, Tomonori Izumitani, Tsutomu Hirao,
Hideki Isozaki, Hideto Kazawa, Eisaku Maeda
NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
{taira,izumi,hirao,isozaki,kazawa,maeda}@cslab.kecl.ntt.co.jp

In TREC-2003, we participated in Question Answering and Genomics Tracks. Since the QA system was essentially the same as the past years' systems[1, 2], we describe our results with the Genomics Track in this paper.

1 Genomics Track Primary Task

Our system consists of two steps. The first step retrieves documents using a keyword search, and the second step scores each document retrieved in the previous step and creates an output file for the TREC submission.

The database provided by TREC consists of more than 500,000 PubMed abstracts. However, less than 50 documents are relevant for most queries. Applying scoring methods to all 500,000 abstracts would create a lot of noise. In the first step, we refined the document set with a simple keyword search.

For the second step, we developed two methods. The first method (Method 1) uses a heuristic scoring system that simply counts the number of verbs and their derived words, which are important to specify the function of a query gene or its product. The second method (Method 2) uses a machine learning technique to score documents.

1.1 Method

1.1.1 Document Retrieval using Keyword Search

TREC provides categories for each query. Namely, official/alias gene/product names, symbols and species. Although species are not necessarily described on documents, "names" or "symbols" should be written in relevant documents. We retrieved all documents that include at least one "name" or "symbol" for each query. They are scored in the next step.

Symbols are represented in various ways in various documents. For example:

- An alias symbol between parentheses follows an official name, such as "p21(Cip)".
- Some symbols are connected by slashes, such as "p21/Waf1/Cip1/Sdi1".
- A combination of the above two cases, such as "p21(WAF/CIP1)".

Additionally, symbols could be written by uppercase characters, lowercase characters or a mixture of both. In this step, we searched for symbols between spaces or marks, such as '-', '/', '(' or ')', without distinction between uppercase and lowercase characters.

8,538 documents for TREC training queries and 18,084 documents for TREC test queries were retrieved in this step.

1.1.2 Method 1 : Heuristic Scoring System

In the previous step, documents that could be relevant to each query gene were obtained. The problem is whether the documents refer to the function of the query gene or a product of it. In this step, all of the retrieved documents are scored for this purpose.

¿From the analysis of all relevant documents for the TREC training data, we found that common verbs or their derived words, such as "express", "bind" or "inhibition",

are often used to describe functions of genes. These words are located adjacent to keywords (query names or symbols). We manually extracted 97 kinds of verbs or their derived words from the vicinity of keywords. We, then, generated a list of words that includes their inflected forms and derived words. Here, we call these words "function words". The list of function words consists of 595 words. The following are parts of this list.

```
bind binds binding bound
control controls controlling controlled
express expresses expressing expressed
expression expressions
indicate indicates indicating indicated
indication indications indicator indicators
...
```

To score each document retrieved in the previous step, a set of words is made using five words before and after the keywords. Then, the score is simply calculated by counting the number of "function words" in the list, allowing for duplication.

1.1.3 Method 2 : Scoring System using SVM

In Method 1, important information for scoring might be lost because of its simplicity and heuristics. We adopted a machine learning techniques to automatically reflect such information to the scoring system.

Machine learning methods such as the Perceptron or Support Vector Machine (SVM) generate discriminant functions whose inputs are mainly vectors and whose outputs are real values. While these methods are usually used as classifiers that output the sign of the discriminant functions, many applications adopt the real value outputs of discriminant functions as confident scores. In this task, we use this value and the SVM as a machine learning method.

Making Vectors from Documents

Representing each document by vector is necessary to make inputs of an SVM [1]. We used the classical "bag of words" model for vectors.

To make vectors, all five words before and after keyword query gene names or symbols are extracted, as well as Method 1. All words except stopwords are used as features of vectors. To decide the values for these vectors, we tried some weighting methods such as TFIDF (term frequency inverse document frequency) and TF in addition to simple binary vectors. However, these weighting methods did not improve the performance. We therefore used binary vectors for all experiments.

Feature Selection

All features of high dimensional vectors are not always effective for discriminant functions. Some features appear in very few documents or have no information for discrimination. The features satisfying the following conditions are eliminated.

- The document frequency is less than Θ_{min}.
- The ratio of positive (relevant) documents to negative (irrelevant) documents is less than Θ_{ratio}.

1.2 Experiment for TREC Training Set

The 8,538 retrieved documents included 233 relevant documents that are 78.5 % of 297 documents provided by TREC[2].

We evaluated Methods 1 and 2 using this data. We divided the data into two sets to create the training and test data for Method 2. The first set is made from queries 1 to 25 and the second is made from the rest. We call the former "Set1" and the latter "Set2". Documents corresponding to queries 21, 35 and 49 were eliminated because they do not include any relevant documents. Set1 consists of 4,675 documents, Set2 consists of 3,560 documents. Set1 and Set2 are used for training and testing, respectively, in Method 2. For Method 2, 12,494 features were extracted from the 4,675 training data.

Table 1 shows the results of Method 1. The method was applied to Set1, Set2 and the whole TREC training set independently, because Method 1 does not need training. The evaluation was performed by mean average precision (MAP) using the "trec_eval"program.

[1] Recently, some methods that calculate values of the discriminant functions directly from character strings or more complicated structures have been developed using kernel methods.

[2] 38 documents in "training-qrels.txt" are not included in the Medline database file, "medline.txt".

Table 1: Mean average precisions of Method 1

Data set	MAP
Set1 (4,675 docs.)	0.250
Set2 (3,560 docs.)	0.322
Whole TREC training set	0.285

Table 2: Mean average precisions of Method 2

Data set	MAP
Set1 (Training set)	0.610
Set2 (Testing set)	0.323
Whole TREC training data	0.573

In Method 2, two kernels, the first and second order polynomial kernels, were applied and various kinds of parameters were examined, namely, Θ_{min} and Θ_{ratio} for feature selection and the SVM soft margin parameter (C). The best parameters, Θ_{min} = 2, Θ_{ratio} = 4 and C = 0.01, were decided by comparing the mean average precision of Set2.

Table 2 shows the results of Method 2. The result for the whole TREC training set was calculated using the TREC training set for SVM training. Set1 and the whole TREC training set have a much higher mean average precision since they are also used for training. Therefore, only the result of Set2 may be an estimation of the TREC test. Methods 1 and 2 yield almost even performances, even though Method 1 utilizes only 595 words in contrast with more than 10,000 words by Method 2. This indicates that verbs and their derived words are crucially important to specify documents that describe the functions of genes or their products.

1.3 Results for Test Set and Discussion

In the first step, 18,084 documents were extracted from the test queries provided by TREC. We applied both Methods 1 and 2 to this data and made two files for submission. Dummy PubMed IDs were filled for queries 7 and 26 because no documents were retrieved in the first step.

Table 3: Mean average precisions for the TREC test set

Method 1	Method 2	Best	Median
0.148	0.153	0.567	0.212

TREC returned average precision scores for each query. The scores of the best, median and worst system were also provided for each query. Table 3 shows the mean average precisions of Method 1 and Method 2 compared with the best and median systems submitted to TREC. The results for Method 1 and Method 2 are almost even, which is consistent with the evaluation for the training set (Subsection 1.2). However, both methods have a little worse than mean average precision.

Figure 1 shows the distribution of the average precisions of Method 1 and Method 2 compared to the best and median systems submitted to TREC. The horizontal axis denotes the average precision and the vertical axis denotes the number of queries. The best scores are significantly high because they do not necessarily come from only one system. The score of the median systems could be a good indicator for average systems. Although the form of distributions are similar among Method 1, Method 2 and the median systems, Methods 1 and 2 have too many low scores less than 0.2. Actually, Method 1 has nine queries whose average precisions are zero and Method 2 has seven queries, of which eight queries are the same for both methods.

This comes from the fact that very few documents were retrieved in the first step. For seven zero score queries, only less than ten documents were retrieved in the first step. Extending queries for the first step considering variations of the description of the gene names, or integrated scoring systems that consider whether a given document describes a query gene and its function simultaneously, will be necessary to improve the performance of our system.

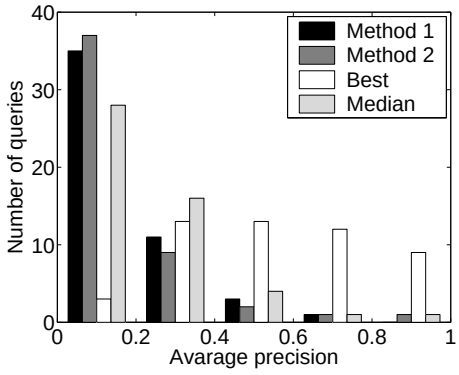

Figure 1: Comparison of Method 1, Method 2, the best systems and the median systems

2 Secondary Track : Automatic Functional Phrases Extraction

We extracted the sequence patterns of the characteristic words (more correctly, the characteristic stems) in the sentences described the gene functions in the training data, in order to generate automatically the phrase that describes the function of a gene.

Next, we scored the test sentences using the information criteria of the sequence patterns.

Last, we output the sentence with the highest score as the phrase explaining the gene's function.

2.1 Labeling Positive and Negative Labels to Training Set

First, as preparation for calculating the information criteria, we gave positive or negative labels to the training sentences according to whether their sentences are close to a correct answer or not. After we divided articles into sentences by our sentence boundary detector, we selected sentences with a small Edit Distance to the actual GeneRIF used as correct answers out of the training set. We gave these positive labels and gave the others negative labels.

More precisely, we labeled sentences whose Edit Distances were 30 % or less than the length of the GeneRIF and the sentence with the smallest Edit Distance as positive. We labeled the other sentences as negative.

2.2 Specification of Gene Name

The information about whether a sentence includes gene names is important for judging whether the sentence includes descriptions about a target gene. We, therefore, replaced the query gene name to "<QUERY_GENE>" tag, and the other gene names to "<SUBSTANCE>" tag.

Although various methods for extracting gene names have already been proposed, these methods need a lot of training data. Therefore, we used the following techniques.

We used gene names and abbreviated gene names registered in the LocusLink and GOA database [3] for searching gene names.

Moreover, we applied the following experiential rules to determine gene names and abbreviated gene names.

- words that are constructed from 3 to 8 characters and are not DNA base pair sequences.

Next, we detected word sequences not satisfied with the following condition in the word sequences that begin with 'the' and end with '(consonant)+ase', '(consonant)+in', '-tor' or '-ssor' as gene names, except for the following case.

- containing Stopwords (Stopwords at PubMed[4] and our original stopwords).
- containing '-ing', '-ed', '.', ';', etc.
- containing only one parenthesis, '(' or ')'.

2.3 Stemming Process

Pattern extraction is possible also from the surface word sequence; however, in the case of, for example, "inhibition of A" and " inhibitor A", these phrases will be treated as different phrases.

In order to avoid this, we extracted stem patterns after stemming to the word using the Porter stemmer [7].

For example, the following sentence,

[3] http://www.ebi.ac.uk/GOA/

[4] http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html#Stopwords

Regulation of Fas-associated death domain interactions by the death effector domain identified by a modified reverse two-hybrid screen.

is stemmed to the following stem sequence.

<regul> <fas-associ> <death> <domain> <interact> <SUBSTANCE> <domain> <identifi> <modifi> <revers> <two-hybrid> <screen>

2.4 Pattern Extraction with Tidal PrefixSpan

We utilized a hyper geometry distribution score (hgs) for extracting stem sequence patterns that appear exclusively to positive examples.

Hisamitsu et al. [3] have proposed a method of weighting words by which the given document set is characterized using an hgs. They showed that words selected by the hgs are effective for standing for the contents of artciles compared with TF-IDF, etc.

Here, a definition of the score using this super-geometry distribution (hgs) is the probability that more than y samples are positive, when x samples are taken without duplication out of the sample set of n containing positive samples of m.
We used $-\log$ (hgs) as statistical criteria.

We extracted patterns using the Tidal PrefixSpan [4] for improvement of speed. PrefixSpan [6, 5] is a high-speed extraction method that can extract high-frequency appearance patterns allowed skips that was proposed by Pei et al.

For example, from the following sentences,

1. I should point out that we need ...
2. I must point out that it is important ...,

PrefixSpan can extract the pattern

"I"-"point"-"out"-"that"

at a high speed.

However, since the original PrefixSpan only takes out high frequency patterns, it is necessary for it to be devised to take out the pattern with high information criteria. Here, we can utilize Tidal SMP (Tidal Statistical Metric Pruning) [8]. Tidal SMP is a technique to accelerate counting the number of patterns with an information criteria.

We used Tidal PrefixSpan, which is a technique of applied Tidal SMP to PrefixSpan, for finding significant patterns with statistically meaning. We used the value of $-\log(hgs)$ divided by the pattern length $(= 1, 2, 3, ...)$ as statistical criteria and scoring points.

2.5 Functional Phrase Output

We scored all the sentences that included test articles by summing up stem pattern scores. Next, we extracted the sentence with a high score for every part (title, abstract, body and caption parts) of the article. Then, we finally selected the output sentence from four sentences by re-scoring with weight. Output sentences are basicaly one sentence. If the sentence was long, we outputed a head part of less than 256 characters of the sentence.

2.6 Experimental Result

We scored patterns with a length of three or less and a frequency of two or more in the training data. We then extracted the top 800 patterns with high hgs values using the Tidal PrefixSpan.

Stem patterns that appear two or more times extracted by Tidal PrefixSpan are shown in Table 4.

<crystallin> (crystallin), <len> (lens), etc., which seldom generally appear, were extracted from the training set. This is because patterns with low frequency may often get a high value of ($-\log(hgs)$ / pattern length).

We show the patterns extracted with a higher rank in Table 5 that appear 100 or more times. This indicates that our method can extract patterns that are likely to appear also in the test data and which are generalized. This shows that the generalized patterns can be extracted with the combination of the cut-off point by frequency and the value using the hyper geometry distribution.

We evaluated the output results by four improvement Dice coefficients. By average of a total of 139 questions, their scores are CD (Classical Dice) : 48.78%, MUD (Modified Unigram Dice) : 50.39%, BD (Bigram Dice) : 31.49% and BP (Bigram Phrase) : 33.79%.

Part of the concrete results is shown in Table 6. This is the result of the higher 1, 5, 10, 50 and 100 ranked when

Table 4: Extracted Stem Patterns (higher 30 pattern, existing more than one frequency).

Pattern	Pos. frq.	Neg. freq.	$-\log(hgs)$ / pattern length
<crystallin>	13	28	44.40
<regul>	31	1095	29.25
<crystallin> <gene>	13	8	27.85
<len>	7	25	21.15
<crystallin> <express>	10	7	21.15
<human>	27	1264	19.45
<signal>	26	1180	19.37
<gene>	33	1887	18.76
<QUERY_GENE>	50	3818	18.71
<SUBSTANCE> <crystallin>	9	10	17.75
<crystallin> <gene> <express>	10	4	15.08
<pathwai>	19	826	15.01
<regul> <SUBSTANCE>	22	511	14.34
<recognit>	7	83	14.09
<SUBSTANCE>	135	18437	13.99
<suggest>	20	1022	13.29
<suffici>	8	139	13.20
<conclud>	6	61	13.08
<gene> <len>	5	0	13.00
<express>	46	4081	12.86
<SUBSTANCE> <crystallin> <gene>	8	4	11.82
<moieti>	4	19	11.78
<co-activ>	5	46	11.53
<pyrophosph>	4	21	11.43
<crystallin> <crystallin>	5	3	10.99
<gtp-bound>	4	27	10.55
<necessari> <gtp-bound>	4	0	10.39
<level> <crystallin>	4	0	10.39
<human> <moieti>	4	0	10.39
<gene> <crystallin>	4	0	10.39

sorting with the results of the Classic Dice coefficient in 139 questions.

Even if the output is apparently close to the correct answer, for example, the 50th problem, a low score can be obtained, because predicted phrases are evaluated only until bi-gram.

These evaluation methods are also a future work.

3 Conclusion

In this paper, we showed characteristic word sequences allowed skips are effective for extracting sentences that described the function of genes in medical documents and showed that scoring by the characteristic word sequence that allows the skip is effective.

Moreover, we showed that the characteristic word sequence that allows the skip can be extracted by Tidal PrefixSpan at a high speed.

Concerning the secondary track, improvement of the evaluation method is greatly required for grasping the

Table 5: Extracted Stem Patterns (higher 30 pattern, existing 100 or more than frequency).

Pattern	Pos. frq.	Neg. freq.	$-\log(hgs)$ / pattern length
<regul>	31	1095	29.25
<human>	27	1264	19.45
<signal>	26	1180	19.37
<gene>	33	1887	18.76
<QUERYGENE>	50	3818	18.71
<pathwai>	19	826	15.01
<regul> <SUBSTANCE>	22	511	14.34
<SUBSTANCE>	135	18437	13.99
<suggest>	20	1022	13.29
<suffici>	8	139	13.20
<express>	46	4081	12.86
<evid>	9	273	10.35
<function>	17	988	9.86
<gene> <express>	16	419	9.73
<regul> <cell>	12	208	9.60
<role>	15	835	9.30
<provid>	9	321	9.15
<transcript>	19	1267	9.09
<drosophila>	6	147	8.39
<necessari>	6	153	8.18
<novel>	6	153	8.18
<cancer>	8	294	8.07
<interact>	17	1177	7.82
<modul>	7	238	7.65
<taken>	5	110	7.64
<SUBSTANCE> <regul>	15	501	7.62
<SUBSTANCE> <SUBSTANCE>	82	8888	7.56
<essenti>	7	244	7.51
<SUBSTANCE> <express>	30	1901	7.45
<high>	9	408	7.43

deeper meaning of sentences.

References

[1] H. Kazawa, H. Isozaki and E. Maeda. NTT question answering system in TREC 2001. *Proc. of The Tenth Text REtrieval Conference (TREC2001)*, 2001.

[2] H. Kazawa, T. Hirao, H. Isozaki and E. Maeda. A machine learning approach for QA and novelty tracks: NTT system description. *Proc. of The Eleventh Text REtrieval Conference (TREC2002)*, 2002.

[3] T. Hisamitsu and Y. Niwa. Topic word selection using a method of word weighting based on conbinatorial probability. *IPSJ SIG-NL, 2000-NL-140 (in Japanese)*, pages 85–90, 2002.

[4] H. Isozaki, T. Hirao and J. Suzuki. On selection criteria of combinatorial features for machine learning. *IPSJ SIG-NL, 2003-NL-158 (in Japanese)*, 2003.

[5] T. Kudo, K. Yamamoto, Y. Tsuboi and Y. Matsumoto. Text mining using linguistic information. *IPSJ SIG-NLP, 2002-NL-148 (in Japanese)*, pages 65–72, 2002.

Table 6: Extracted higher 1, 5, 10, 50, 100th rank phrases.

Rank 1st(TREC ID 9 : **CD:100.00%**, **MUD:100.00%**, **BD:100.00%**, **BP:100.00%**)
Answer : Regulation of intracellular pH mediates Bax activation in HeLa cells treated with staurosporine or tumor necrosis factor-alpha
Predicted : Regulation of Intracellular pH Mediates Bax Activation in HeLa Cells Treated with Staurosporine or Tumor Necrosis Factor-alpha

Rank 5th (TREC ID 10 : **CD:100.00%**, **MUD:100.00%**, **BD:100.00%**, **BP:100.00%**
Answer : Apocytochrome c blocks caspase-9 activation and Bax induced apoptosis
Predicted : Apocytochrome c Blocks Caspase-9 Activation and Bax-induced Apoptosis

Rank 10th (TREC ID 90 : **CD:96.55%**, **MUD:95.24%**, **BD:94.74%**, **BP:92.31%**
Answer : Activity in the nucleus accumbens shell controls gating of behavioral responses to emotional stimuli.
Predicted : CREB activity in the nucleus accumbens shell controls gating of behavioral responses to emotional stimuli

Rank 50th (TREC ID 69**CD:51.28%**, **MUD:48.00%**, **BD:17.39%**, **BP:14.29%**
Answer : there is a mechanically coupled transcriptional circuit that promotes binding of p38 to Sp1 in the nucleus
Predicted : Interaction of p38 and Sp1 in a Mechanical Force-induced, beta1 Integrin-mediated Transcriptional Circuit That Regulates the Actin-binding Protein Filamin-A

Rank 100th (TREC ID 33 :**CD:31.82%**, **MUD:32.26%**, **BD:13.79%**, **BP:23.53%**
Answer : the JH2 domain contributes to both the uninduced and ligand-induced Jak-receptor complex, where it acts as a cytokine-inducible switch to regulate signal transduction
‘redicted : The Pseudokinase Domain Is Required for Suppression of Basal Activity of Jak2 and Jak3 Tyrosine Kinases and for Cytokine-inducible Activation of Signal Transduction

[6] J. Pei, J. Han, B. Mortazavi-Asl, and H. Pinto. Prefixspan: Mining sequential patterns efficiency by prefix-projected pattern growth. In *Proc. of the International Conference on Data Engineering (ICDE)*, pages 215–224, 2001.

[7] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[8] J. Sese and S. Morishita. Answering the most correlated n association rules efficiently. In *Proc. of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 410–422, 2002.