An information extraction method for multiple data sources

Hironori Mizuguchi and Dai Kusui

Information and Media Processing Laboratories, NEC Corporation
8916-47 Takayama-cho Ikomashi, Nara 631-0027, Japan
+81-743-72-3680

{hironori@ab, kusui@ct} .jp.nec.com

ABSTRACT

We developed a method of information extraction for multiple data sources or for various kinds of datasets like Internet web pages. Generally, because many different writing styles or vocabularies exist among different kinds of data, the accuracy of information extraction using various kinds of datasets is not better than that using a single kind of data. Our method divides the data by clustering and learns extraction rules to increase accuracy even if we use various kinds of datasets. In our experiment, we applied our method to a NTCIR8 Technical Trend Map Creation subtask that uses two kinds of data, patents and technical papers, and obtained the better precision than normal information extraction method.

Categories and Subject Descriptors

I.2.7 [**ARTIFICIAL INTELLIGENCE**]: Natural Language Processing – *Text analysis*; H.3.1 [**INFORMATION STORAGE AND RETRIEVAL**]: Content Analysis and Indexing – *Linguistic processing*

General Terms

Algorithms, Experimentation

Keywords

Entity extraction, Clustering, Machine learning

1. INTRODUCTION

Recently, effective uses are expected of a great deal of text information such as Web pages and company documents. In these texts, many words have specific meanings (called semantic classes) such as a person's name, a location's name, an organization's name, and so on. By extracting these words, we can effectively use the text information on a Q and A system and for text categorization, machine translation, and so on. For example, we can automatically create a technical trend map with two axes for technologies and effects by extracting technology names and their effects from patents and technical papers. By looking at the trend map, we can recognize what technology exists and how it is related to which effect.

Named Entity Resolution extracts words that have semantic classes. Yamada's very popular method [7] creates rules that recognize such words by a machine learning technique that uses training data annotated with semantic classes. This method is divided into two phases: learning and applying. In the learning phase, based on the assumption that identical semantic class words have similar neighbor words, machine learning creates rules that sort a word into positive and negative using the annotated words and the neighbor words as a positive example and the un-annotated words and neighbors as a negative example. In the apply phase, the method decides whether an input word is a semantic class by using the rules.

This method is commonly used with a single kind of data in which the neighbor words are similar because the writing style of the same type of data is similar. For neighbor words that are not similar, prior method cannot create good rules and cannot get good accuracy. Table 1 shows an example of prior method with multiple kinds of data (patents and technical papers) in a Technical Trend Map Creation subtask in NTCIT8. The first column denotes the kinds of training datasets: only paper, only patents, or a combination. The second column denotes the kind of testing dataset. The rest of the columns show the macro-averages of the extraction semantic classes, TECHNOLOGY, ATTRIBUTE, and VALUE. The precision, recall, and f-value of the mixed data in the training data are lower than those of the single kinds.

In recent years, the utilization of information that contains such various kinds of data as internet web pages or intranet enterprise documents is expected to increase. In the Technical Trend Map Creation subtask of NTCIR8, the kind of each data was already given. However, to use internet or intranet data, we do not know the kind of each piece of data or how the dataset should be separated.

We researched an information extracting method that gets good accuracy even if the data contains various kinds of data. Our clustered learning method makes clusters from training data and rules from each clustered training data using machine learning techniques. Our method can divide problem spaces like kinds of data by clustering and creates rules for each problem space. However, the clustered training data by only clustering has bias: the amount of training data in each cluster and the density in each problem space. To limit these effects, our method modifies the data of clusters. Specifically, the data in large and high density clusters are moved to another cluster.

In this paper, first, we describe an abstract of the Technical Trend Map Creation subtask of NTCIR8. Next, we explain our clustered learning method and show experiment and subtask results. Finally, we describe the discussions.

Table 1. Comparison accuracy between single and multiple kinds of training data

Training	Test	Precision	Recall	F
Paper	Paper	51.97	23.64	32.40
Paper + Patent	Paper	45.62	18.07	25.75
Patent	Patent	66.24	40.68	50.35
Paper + Patent	Patent	63.54	36.65	45.95

2. TECHNICAL TREND MAP CREATION SUBTASK

The purpose of this task is extracting technology names and the effects that use the axes of a technical trend map from technical papers and patents [4]. The following are the practical semantic classes:

- TECHNOLOGY: algorithms, tools, materials, or data used in each paper or patent
- EFFECT: ATTRIBUTE and VALUE pairs
- ATTRIBUTE: an attribute in the effect of a technology
- VALUE: values related to attributes in an effect

A task organizer distributes the training dataset, and participants develop systems that extract these semantic classes.

装置は、<TECHNOLOGY>パリティデータ</TECHNOLOGY>を持つことにより<EFFECT><ATTRIBUTE>一定数の記憶媒体の障害</ATTRIBUTE>を<VALUE>修復</VALUE></EFFECT>することが可能である。

Figure 1. Sample data of Technical Trend Map Creation subtask

3. CLUSTERED LEARNING METHOD

Our clustered learning method recognizes text's semantic classes by using rules learned by machine learning techniques. It divides training data into clusters and makes rules from each clustered training data. However, clustered training data have some bias: the amount of training data in each cluster and the density in each problem space. To limit these effects, our method modifies the data of clusters.

Our method divides problem spaces like data kinds by clustering and creates rules of each problem space. Therefore, for training data that contain various kinds of data, our method can get good accuracy.

Figure 2 shows an overview of our clustered learning method that is divided into two phases, learning and applying. In the learning phase, the training data is divided into clusters. Next, the clustered training data are modified to limit biases. Then, rules are made from each clustered training data. In the applying phase, a specific cluster is selected that resembles test data as an input. Rules related to the selected clusters are applied to recognize each word's semantic class.

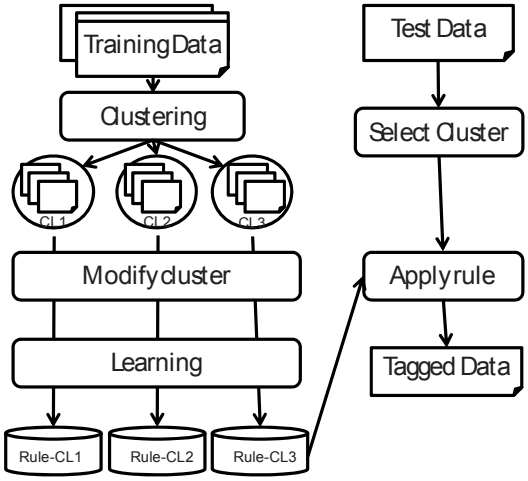

Figure 2. Overview of clustered learning method

3.1 Clustering

Our method divides training datasets into clusters by a clustering technique. A clustering target is a document or the context. The context means the information of each word and its neighbor words. For document clustering, the clusters denote kinds of data. For context clustering, the clusters denote kinds of words.

Document clustering makes clusters from each document in the training datasets. Each document is represented as a document vector that consists of the word whose part of speech is noun, verb, adjective, adjectival verb or adverbs and the number of words in the document.

Context clustering makes clusters from each appearance of all words in the documents in the training dataset. Each appearance of words is represented as a feature vector that consists of the word's grammatical information and its neighbor words. These features are mentioned below in this section.

Many clustering methods are available, including k-means [6], pLSI [3], and so on. K-means with cosine similarity is suitable for document clustering because each document vector has a lower dimension number than the feature vector's one in content clustering. pLSI is better for context clustering because it can reduce the dimensions of vectors; pLSI can handle high-dimensional data like feature vectors.

3.2 Modification of clusters

The purpose of the modification of clustered data is to limit the effects of biases and to generalize each problem space defined by each cluster. The biases of the clustered training dataset are the amount of training data in each cluster and the density in each problem space. These biases, which harm machine learning, and their effects are described in more detail as follows.

Figure 3 shows a conceptual figure of modification clusters. The small circles, the big circles, and the x marks denote the vectors related to each piece of data, the clusters, and the centers of each cluster, respectively. First we considered the clustered training data, as shown on the right side. In this case, since most vectors in each cluster are close to the center of each cluster, our method cannot recognize a word located outside of these clusters, so a set of rules in these clusters does not have good generalization performance. Moreover, the amount of data in cluster c is smaller than the other clusters. Thus, each rule in each cluster has different performance.

To limit the effects from these biases, our method modifies the data in the clusters by moving them among clusters. Such modification averages the amount and the density of each cluster.

First, AmountBias and SimBias are calculated as follows:

$$AmountBias(C_i) = \frac{n_i}{n}$$

$$SimBias(C_i) = \frac{\frac{1}{n}\sum_{j=1}^{n_i}(1 - sim(x_{ci}, x_{ij}))^2}{\frac{1}{N}\sum_{k=1}^{k}\sum_{j=1}^{n_i}(1 - sim(x_{ck}, x_{kj}))^2}$$

In the equation, C_i is the i[th] cluster, n_i is the number of vectors in the i[th] cluster, and $\bar{n}$ is the average of the number of vectors in each cluster. Therefore, AmountBias is the ratio between the number of vectors in C_i and the average. x_{ci} is a center vector of C_i. x_{ij} is an j[th] vector in C_i. $sim(x, y)$ is the similarity between x and y. Therefore, SimBias is the ratio between the mean square of the dissimilarity between the center and each vector.

A cluster is selected whose AmountBias is higher than a particular threshold and whose SimBias is lower than a particular threshold. The vector in the selected cluster is moved to another cluster whose center vector resembles the vector.

These steps are repeated until all clusters satisfy the AmountBias and SimBias thresholds.

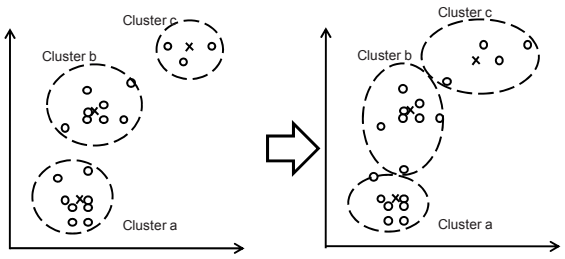

Figure 3. Modification of clusters

3.3 Learning rules

By using each bit of clustered training data, machine learning makes recognition rules. Each rule is made from each cluster. The learner creates rules that sort a word into positive and negative using the annotated word and the context as a positive example and the un-annotated word and the context as a negative example. The context contains the features of the target word, its two neighbor words, and words that have a modification relationship of the target and neighbor words. Practical features include:

- character string
- plain form
- part of speech
- semantic label from language analysis

Additionally, there are some practical rule creation methods, such as rules that only recognize the semantic class of each word or recognize the semantic class and its range like the BIO method [7]. We can choose any method.

3.4 Applying phase

First, a cluster is selected that contains rules applied to input words. The similarity between the inputted document or context and the centers in each cluster is calculated. Then the cluster is selected with the highest similarity. If the cluster is made from document vectors at the learning phase, then the similarity between the documents is calculated. If the cluster is made from context vectors at the learning phase, then the similarity between contexts is calculated. If the score is 0, our method does not apply rules because we cannot select clusters.

Then the rules, which are related to the selected cluster, are applied to the inputted word.

4. EXPERIMENTS AND RESULTS

To evaluate our method, we did experiments using the Technical Trend Map Creation subtask data. First, we show the result of a formal run. Next, we represent the effects of clustering. Due to time limitations, we couldn't experiment on the modification of clusters. We want to represent the effect of the modification of clusters near future.

Before describing the formal run results, we show some conditions.

We used bisecting K-means [6] as the only clustering method and CLUTO [2] for implementation. A cluster's center is medoid calculated by CLUTO as the highest z-score. A vector's z-score is the difference between the average similarity of the vector in its cluster and the averaged similarities of all vectors in the same cluster.

The rule creation method makes rules that recognize whether it is a semantic class. Rules cannot recognize the range of semantic classes like BIO tags. We make four rules in each cluster for each semantic class: TECHNOLOGY, ATTRIBUTE, VALUE, and OTHER. If a semantic class expression consists of multiple words, rules can only recognize its last word.

SVM is the machine learning technique, and we use libSVM [1] for its implementation with linear kernel and default parameters.

Language analysis was done by Jana [5], which is developed and studied by the NEC corp. .

4.1 Experiment 1: Formal run

We applied our method to a Technical Trend Map Creation subtask. Due to time limitations, we couldn't use clustering and modification.

Additionally, rules can only recognize the last word of the semantic classes of TECHNOLOGY, ATTRIBUTE, and VALUE. Therefore, we have to take a chunk from each semantic class word and make an EFFECT class. We make a chunk of TECHNOLOGY and ATTRIBUTE as a sequence of the modification relation words of the last word. We make an EFFECT class as a pair of an ATTRIBUTE class and a VALUE class that are located within two words.

Table 2 shows our result of a formal run (In [4], our run id is 'ONT').

Table 2. Results of formal run

KIND	TAG	RECALL	PRECISION	F
PAPER	TECHNOLOGY	9.1	21.9	12.9
	ATTRIBUTE	8.1	15.4	10.6
	VALUE	12.2	26.7	16.8
	EFFECT	2.7	18.2	4.7
PATENT	TECHNOLOGY	4.7	8.0	6.0
	ATTRIBUTE	21.9	29.6	25.2
	VALUE	33.8	50.3	40.4
	EFFECT	12.5	33.9	18.2

4.2 Experiment 2: Effect of clustering

Figure 4 shows the precisions and recalls of the document and context clustering of the mixed training data in the learning phase. In Figure 4, each line denotes different test data and different clustering method. For example, 'MIX_PATENT_DOC' shows results of PATENT test data and document clustering with mixed training data. The precision and recall are the macro-averages of TECHNOLOGY, ATTRIBUTE, and VALUE by 10-fold cross-validation.

Additionally, we measured whether the last word of each class is correct. Since we can compare the single and combined kinds, the training dataset includes 500 documents: 250 from papers and 250 from patents.

Precision

- Document clustering (MIX_PATENT_DOC, MIX_PAPER_DOC): when the number of clusters is two, the precision outperformed the normal (0 clusters). The precision of paper test data with two document clusters showed a dramatic increase as compared with The precision of paper test data without document clustering.
- Context clustering (MIX_PATENT_CON, MIX_PAPER_CON): There were few changes.

Recall

- The recalls of all tests decreased by increasing the number of clusters. The rules became more specific by increasing the clusters.

Table 3 shows the results of the single kind (patent or paper) training data, the mixed training data with document clustering (two clusters), and the mixed training data without clustering. The precision of mixed training data with clustering and patent or paper test data is higher than the precision of mixed training data without clustering and patent or paper test data, respectively. Especially, the precision of mixed training data with clustering and paper test data is higher than the precision of paper training data and paper test data. But recall is down.

Table 3. Comparison of each method (single kind (PATENT or PAPER), MIX(CLST=2) that use clustered learning method with two document clusters and MIX that uses normal mix training data)

TRAINING	TEST	PREC	RECALL	F
PATENT	PATENT	66.24%	40.68%	50.35%
MIX(CLST=2)	PATENT	63.89%	33.80%	43.75%
MIX	PATENT	63.54%	36.65%	45.95%
PAPER	PAPER	51.97%	23.64%	32.40%
MIX(CLST=2)	PAPER	52.67%	15.42%	23.74%
MIX	PAPER	45.62%	18.07%	25.75%

5. DISCUSSIONS

5.1 Cluster and the effect

Next we discuss the results of document and context clustering.

The document clustering results in the training phase are shown at Table 4 with the number of clusters (2, 3, 4), the cluster id, and the number of documents for each kind in the cluster. We got satisfactory results that met our expectations because we sorted the training documents into different kinds. This is one of the factors of good effects.

The context clustering results in the training phase are shown in Table 5 with the number of contexts in each cluster and in Table 6 with a representative example of contexts in each cluster. There is no consistency of contexts in the same cluster. This is one of the reasons of poor results in context clustering in Figure 4.

One cause of this result is the clustering method (K-means). The number of dimensions of the feature vector used by context clustering is huge (760,000). Therefore, we should use methods that can compress huge dimensions, like pLSA.

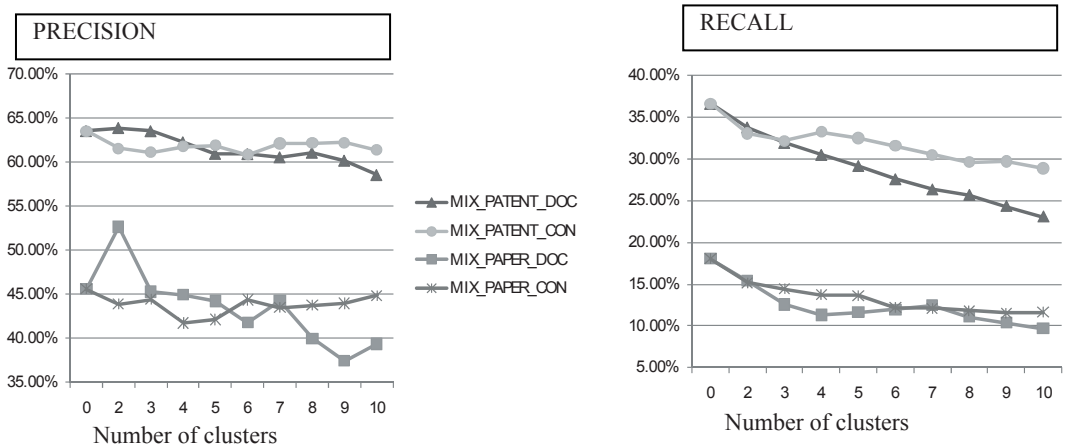

Figure 4. Result of clustered learning method in all clusters

Table 4. Amount of data in each document cluster

CLST	ID	PAPER	PATENT	SUM
2	0	5	**209**	214
	1	**173**	2	175
3	0	**79**	1	80
	1	**94**	1	95
	2	5	**209**	214
4	0	3	**124**	127
	1	2	**85**	87
	2	**79**	1	80
	3	**94**	1	95

Table 5. Amount of data in each context cluster

CLST	ID	PAPER	PATENT	SUM
2	0	8322	19370	27692
	1	8137	16683	24820
3	0	1852	3295	5147
	1	8322	19370	27692
	2	6285	13388	19673
4	0	1852	3295	5147
	1	3218	7472	10690
	2	5104	11898	17002
	3	6285	13388	19673

Table 6. Example context in each context cluster

CLST	ID	CONTEXT
2	0	制御を禁止したり振れ 電位に応答して導通状態と 本発明によると安価な
	1	設定する構成により 第1及び第2の記憶 なる。
3	0	図る。 至る。 なる。
	1	制御を禁止したり振れ 電位に応答して導通状態と 本発明によると安価
	2	柔軟な組合せの方法 設定する構成により 印刷機構部に依頼できる形式に
4	0	図る。 至る。 なる。
	1	制御を禁止したり振れ 小型化を達成することができる。 線にもなる。
	2	動作の回数にたいし 発明の効果】 端子点の低下
	3	設定する構成により 第1及び第2の記憶 なる。

5.2 Amount of training data in each kind

Table 7 shows the number of words and the frequencies of all words in each kind of training dataset in Experiment 2. The patent's training data is bigger than the paper's training data because its frequency per word is bigger. The normal mixed method is lower than our method because the patent effect is bigger than the paper effect. Since our method can divide training data into different kinds, the precision of paper is especially good.

Table 7. Number of words and frequency of all words in each kind of data

CLASS		PAPER	PATENT
ATTR	WORDS	227	384
	FREQ	288	613
TECH	WORDS	251	300
	FREQ	422	899
VALUE	WORDS	179	239
	FREQ	279	595
OTHER	WORDS	5094	5701
	FREQ	17234	38491

5.3 Selection of cluster at applying phase

First, we discuss the effect of the similarity threshold used by selecting the cluster phase. Figure 5 shows the precision and recall of document clustering at each similarity threshold with two clusters. The precision increases until the threshold reaches 0.15. This means that the data near the center of each cluster are correct. However, when the threshold exceeds 0.15, the precision and recall decrease because our method cannot select a cluster. How to decide a good threshold is a problem.

Our method selects a cluster that has the highest similarity between its center and the input vector. However, since the amounts of training data and the density of each cluster are different, we should determine a cluster with different thresholds of each cluster or a different way that is not only based on similarity.

5.4 Effects of clustered learning method

Our method outperformed prior method when the number of cluster was applicable, because the clusters of document clustering are divided into the kinds of data. From Table 3, when the number of clusters is two, the precision outperformed the normal (0 clusters). From Table 4, document clustering sorted the training documents into different kinds. If we use various kinds of data like internet web pages and intranet enterprise documents, our method can increase the accuracy of information extraction.

In my opinion, the reason of goodness of the paper's precision at the two clusters in Figure 4, the training data of technical papers is small. The difference of training data amount between patents and papers is led from Table 7. In prior method (normal mixture), the effect of patents whose training data is big is stronger than the paper one. So, the accuracy of paper is low. However our method can divide training data into their kinds and can deny this bad effect.

Because the rules became more specific by increasing the clusters, the recall decreased by increasing the number of clus-

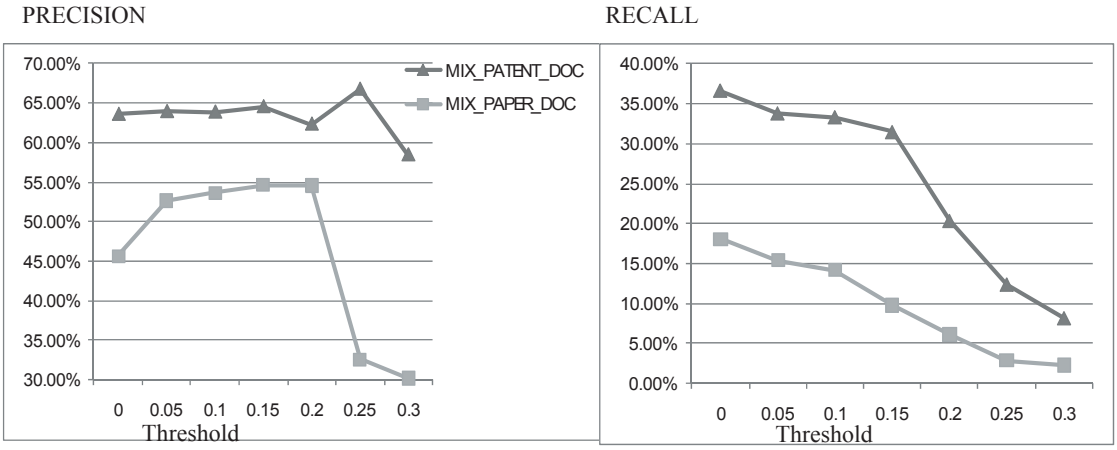

Figure 5. Results in each threshold of similarity between test data and centre of a cluster

ters. We have to decide an applicable cluster size. It can be testing with training data.

Context clustering could not get good result. We think its cause was the problem of clustering method and cluster size. The clustering methods that can compress huge dimensions, like pLSA, is better than K-means because the number of dimensions of the feature vector used by context clustering is huge (760,000). Cluster size should be bigger than 10 because kinds of contents is more various than documents. We have done a simplified experiment when the number of cluster was 1000. We tested micro-averages of ATTRIBUTE, TECHNOLOGY and VALUE at nine cluster's rules of 1000 clusters only. The precision of patents and papers are 76.68 and 53.85, respectively. These precisions are as same as or higher than the results of a single kind in Table 3. We want to do experiments of other clustering methods and big cluster size.

6. CONCLUSION

We proposed an information extraction method called clustered learning for various kinds of data and applied it to a NTCIR8 Technical Trend Map Creation subtask. Even if data contains multiple kinds of data, our method obtained similar results to a single one when training data can be divided into their kinds of data.

Future work includes experiments with the modification of clusters and recall effects. We want to use another clustering method like pLSA in context clustering. Another way of deciding select clustering will also be studied.

7. References

[1] Chang, C.–C. and Lin, C.-J. 2001. libSVM: a library for support vector machines. (Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm)

[2] CLUTO, 2006, CLUTO version 2.1.2a, Software Package for Clustering High-Dimensional Datasets", March 2010, http://glaros.dtc.umn.edu/gkhome/views/cluto.

[3] Hofmann, T. 1999. Probabilistic latent semantic indexing. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99), pp.50-57

[4] Nanba, H., Fujii, A., Iwayama, M. and Hashimoto, T. 2010. Overview of the Patent Mining Task at the NTCIR-8 Workshop. Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access.

[5] Sakao, Y., Ikeda, T., Satoh, K. and Akamine, S. 2005. Japanese language analysis for syntactic tree mining to extract characteristic contents. Proceedings of The Tenth Machine Translation Summit (MT Summit X), pp 339-345.

[6] Steinbach, M., Karypis, G. and Kumar, V. 2000. A comparison of Document Clustering Techniques. KDD Workshop on Text Mining (KDD2000), Boston.

[7] Yamada, H., Kudo, T. and Matsumoto, Y. 2002. Japanese Named Entity Extraction Using Support Vector Machine. Transactions of Information Processing Society of Japan 43(1), pp.44-53.[in Japanese]