Paper Classification for Recommendation on Research Support System Papits

Tadachika Ozono,[†] *and Toramatsu Shintani*[††],

Computer Science and Engineering, Graduate School of Engineering
Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555 JAPAN

Summary

We have developed a research support system, called Papits, that shares research information, such as PDF files of research papers, in computers on the network and classifies the information into types of research fields. Users of Papits can share various research information and survey the corpora of their particular fields of research. In order to realize Papits, we need to design a mechanism for identifying what words are best suited to classify documents in predefined classes. Further we have to consider classification in cases where we must classify documents into multivalued fields and where there is insufficient data for classification. In this paper, we present an implementation method of automatic classification based on a text classification technique for Papits. We also propose a new method for using feature selection to classify documents that are represented by a bag-of-words into a multivalued category. Our method transforms the multivalued category into a binary category to easily identify the characteristic words to classify category in a few training data. Our experimental result indicates that our method can effectively classify documents in Papits..

Key words:

Knowledge Management, Recommendation, Text Categorization, Feature Selection.

1. Introduction

We have developed a research support system, called Papits [2][8]. Papits has several functions that allow it to manage research information, i.e., a paper sharing function, a paper classifier, a paper recommender, a paper retriever, and a research diary. The paper sharing function facilitates to share research information, such as the PDF files of research papers, and to collect papers from Web sites. The function of automatic classification can classify research information into several research fields. This function enables users to search papers based on category of their interest. Automatic classification in Papits has a structure that gradually improves accuracy through feedback from users. In this paper, we mainly discuss paper classification.

In automatic text classification, one of the main problems is how to identify what words are best suited to classify documents in predefined classes. Feature selection techniques are therefore needed to identify these words, and one such technique uses the information gain (IG) metric[9] assessed over the set of all words encountered in all texts[6][11]. Soucy [11] proposed a feature selection method based on IG and a new algorithm that selects features according to their average cooccurrence. It yielded good results on binary class problems. Automatic classification in Papits needs to classify documents to be classified into the multivalued category, since researches are organized by several fields. Since there are a lot of research fields, it is hard to collect enough training data. When the number of training data in one category is small, feature selection becomes sensitive to noise and irrelevant data. Further, as previously pointed out, there may not necessarily be enough training data. This paper proposes a feature selection method for classifying documents, which is represented by a bag-of-words, into the multivalued category. It transforms the multivalued category into a binary category, and features are selected using IG.

The remainder of this paper is organized as follows: First, we show an outline of our Papits research support system. Second, we describe classification method and propose the feature selection algorithm for managing research papers. Third, we discuss the experimental results we obtained using our algorithm and prove its usefulness. Fourth, we discuss the functions of Papits. Fifth, we compared our work with related works. Finally, we conclude with a brief summary and discuss future research directions.

2. Research Support System Papits

This section presents an outline of Papits, which is a research support system, implemented as a web application (using WebObjects: Web Objects is a tool for creating a Web Application, developed by Apple). Users can access via a web browser. Papits has several functions that manage research information, i.e., paper sharing, a paper classifier, a paper recommender, a paper retriever, and a research diary. The knowledge management of Papits supports surveys by through these functions. This paper mainly discusses the paper classifier function, which can provide intense support to surveys on fields of research interest. When users want to look for papers they are interested in, they can easily find these by tracing the category or retrieving or using the recommender.

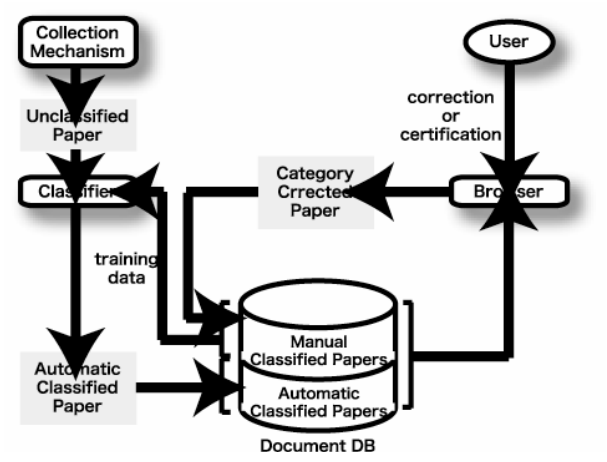
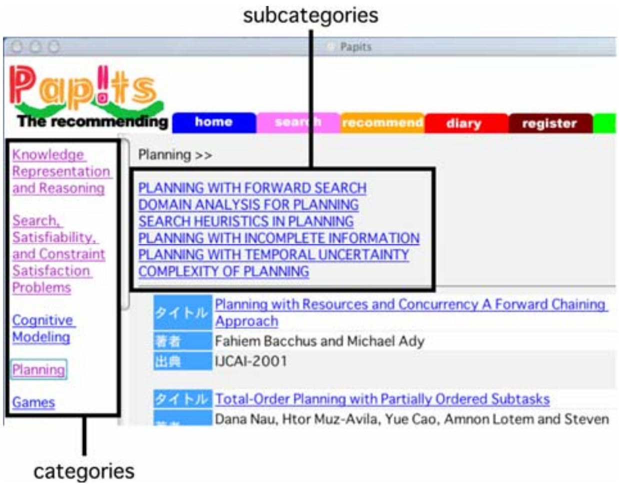


Fig. 1 Work Flow of Classification in Papits.



Knowledge Representation and Reasoning

Search, Satisfiability, and Constraint Satisfaction Problems

Cognitive Modeling

Planning

Games

Planning >>

PLANNING WITH FORWARD SEARCH
DOMAIN ANALYSIS FOR PLANNING
SEARCH HEURISTICS IN PLANNING
PLANNING WITH INCOMPLETE INFORMATION
PLANNING WITH TEMPORAL UNCERTAINTY
COMPLEXITY OF PLANNING

タイトル	Planning with Resources and Concurrency A Forward Chaining Approach
著者	Fahiem Bacchus and Michael Ady
出典	IJCAI-2001
タイトル	Total-Order Planning with Partially Ordered Subtasks
著者	Dana Nau, Htor Muz-Avila, Yue Cao, Amnon Lotem and Steven

Fig. 2 Browsing classified papers.

Figure 1 illustrates the Papits automatic classification process. Papits first collects papers from users, web sites, and the other sources. In this step, the papers have not been yet classified. The unclassified papers are classified by a classifier that uses manually classified papers in the document DB as training data. Here, we have assumed that classification aided by the user is correct, and papers classified by the classifier cannot be guaranteed to be perfectly correct. Papers classified by the classifier are stored in databases as automatic classified papers, and is not used as training data.

While browsing for a paper, if a user corrects or certifies a category for that paper, it is stored as manually classified paper. Training data increases by going through this step, and classification accuracy improves.

Figure 2 has the results obtained for classification in Papits. When a user wants to look for papers of interest, it can be found based on the category of interest. Additionally, users can narrow the range of the field of survey based on subcategories. In this way, users can scrutinize their field of interest through the automatic paper classifier.

3 Automatic Classification

Automatic classification helps users locate papers by following their category of interest. The main problem in automatic text classification is to identify what words are the most suitable to classify documents in predefined classes. This section discusses the text classification method for Papits and our feature selection method.

3.1 Text Classification Algorithm

k-Nearest Neighbor (kNN) and Support Vector Machine (SVM) have frequently been applied to text categorization[12]. Yang describes kNN and SVM are an almost equivalent performance[12]. Section 4 discusses the experimental results using these text classification algorithms.

3.1.1 KNN

The kNN algorithm is quite simple: kNN finds the *k* nearest neighbors of the test document from the training documents. The categories of these nearest neighbors are used to weight the category candidates. The similarity score of each neighbor document to the test document is used as the weight for the categories of the neighbor document. If several *k* nearest neighbors share a category, then the per-neighbor weights of that category are added, and the weighted sum is used as the likelihood score for that category with respect to the test document. By sorting the scores of the candidate category, a ranked list is obtained for the test document.

Typical similarity is measured with a cosine function:

$$\cos(x_1, x_2) = \frac{\sum_{j=1}^{n} a_j(x_1) \cdot a_j(x_2)}{\sqrt{\sum_{j=1}^{n} a_j(x_1)^2 \cdot \sum_{j=1}^{n} a_j(x_2)^2}}$$

where x_1 and x_2 are documents, and x is a document vector $\langle a_1(x), a_2(x), \cdots, a_n(x) \rangle$. $a_j(x)$ is the weight of the *j*-th feature (word) on *x*.

We assumed that the weight of each feature would be the same:

$a_j(x) = 1$: if the $j-$th word is in document x

$a_j(x) = 0$: otherwise

3.1.2 SVM

The formulation of SVM is constructed starting from a simple linear maximum margin classifier [1]. A general linear SVM can be expressed as Eq. 1.

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b \qquad (1)$$

where $f(\mathbf{x})$ is the output of the SVM, $\mathbf{x}$ is the input document, b is a threshold, $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$, $\mathbf{x}_i$ is a stored training document, $y_i \in \{-1,+1\}$ is the desired output of the classifier, and α_i are weights. The margin for this linear classifier is $1/\|w\|$. Hence the training problem is to minimize $\|w\|$ with respect to constraint Eq. 1

The linear formulation cannot classify nonlinearly separable documents. SVMs get around this problem by mapping the sample points into a higher dimensional space using a kernel function. A general non-linear SVM can be expressed as Eq. 2.

$$f(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - b \qquad (2)$$

where K is a kernel function that measures the similarity of stored training documents $\mathbf{x}_i$ to the input $\mathbf{x}$ $y_i \in \{-1,+1\}$ is the desired output of the classifier, b is a threshold, and α_i is weights that blend the different kernels.

The formulation of SVM was based on a two-class problem, hence SVM is basically a binary classifier. Several different schemes can be applied to the basic SVM algorithm to handle the n-category classification problem. One of schemes to handle the n-category is one-versus-rest approach. The one-versus-rest approach works by constructing a set of n binary classifiers for an n-category problem. The k-th classifier is trained with all of the documents in the k-th category with positive labels, and all other documents with negative labels. The final output is the category that corresponds to the classifier with the highest output value.

$$f(\mathbf{x}) = \arg\max_k \sum_{i=1} \alpha_i^k y_i K^k(\mathbf{x}_i, \mathbf{x}) - b$$

4. An Algorithm for Feature Selection

Feature selection techniques are needed to identify the most suitable words to classify documents, and to reduce the computation costs of classifying new documents. In Papits, automatic classification needs to classify documents into the multivalued category, because research is organized in various fields. However, feature selection becomes sensitive to noise and irrelevant data compared to cases with few categories. There may also not be enough registered papers as training data to identify the most suitable words to classify into the multivalued category in Papits. We propose feature selection to classify documents, which is represented by a bag-of-words, into the multivalued category.

Several existing feature selection techniques use some metric to determine the relevance of a term with regard to the classification criterion. IG is often used in text classification in the bag-of-words approach [3][7][10].

$$IG(A,X) = \left(-\sum_{c \in C} \frac{|X_c|}{|X|} \log_2 \frac{|X_c|}{|X|} \right) - \left(-\sum_{v \in Values(A)} \sum_{c \in C} \frac{|X_{c,v}|}{|X|} \log_2 \frac{|X_{c,v}|}{|X_v|} \right)$$

where C is the set of all categories, and each of categories is denoted as c. A is an arbitrary feature (word), v is a value of A, and the set of value of feature A is denoted as $Values(A) = \{0, 1\}$. If feature A is in a document, then value v is 1. Otherwise v is 0. X denotes a set of all documents, and X_c,X_v,$X_{c,v}$ denote sets of documents that are included in category c, taking feature value v, and belonging to category c as well as taking the feature value v. $|X|$ indicate the number of elements of set X.

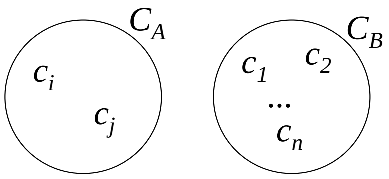


Fig. 3. the multivalued category into the binary category

In this formula, as the number of elements of C increases, documents are divided into more categories. Hence, the IG value becomes sensitive to noise and the irrelevant data. Our method transforms the multivalued category into a binary category, increases the number of data in one category, and does feature selection using IG.

Figure 3 presents the idea behind this method with the set of categories $\{c_1,c_2,\ldots,c_i,\ldots,c_j,\ldots,c_n\}$. If suitable words to classify into documents various combinations of categories are found, since a document category is predicted by a combination of words, we thought it would be possible to classify each category by combining these words. For example, let us suppose the following case:

- set C_A consists of categories c_i and c_j
- set C_B consists of categories other than c_i and c_j
- set C_S consists of categories c_i and c_k
- set C_T consists of categories other than c_i and c_k
- word w_a is suitable to classify C_A and C_B
- word w_b is suitable to classify C_S and C_T

If a combination of w_a and w_b can be found, a classifier can classify original categories c_i, c_j, and c_k. Our feature selection method can be used to locate w_a and w_b .

```
V = set of words, sorting by information gain
    (initial condition = {})
D = set of documents\\
C = set of categories\\
k = arbitrary number of features\\
l = arbitrary number of categories\\
IG$_{\_CA,CB}$ (w,D) : IG of documents D on word w,
                        relative to categories $C_A$ and $C_B$
add(V,w,IG) : word w is added to V sorted by IG value

Feature_Selection_Algorithm()
 for each combination $C_A$ of C choose 1 or 2 categories
    $C_B$ = C - $C_A$
    IGvalue = IG$_{CA,CB}$(w,D)
    if ($max < IGvalue) then
       max = IGvalue
    add(V,w,max)
 return k higher ranks of V.
```

Fig. 4. Proposing Feature Selection Algorithm

Figure 4 shows the proposed feature selection algorithm. First, new category C_A is a set that consists of two or less categories that are selected from a set of categories C, and C_B is a set of elements of C except for categories that constitute C_A . For all combinations of these, IG is assessed over the set of all words encountered in all texts, let the highest value of IG be the importance of word w. IG for new categories { C_A , C_B } is determined by the following:

$$IG_{C_A,C_B}(A,X) = \left(\frac{|X_{C_A}|}{|X|}\log_2\frac{|X_{C_A}|}{|X|} + \frac{|X_{C_B}|}{|X|}\log_2\frac{|X_{C_B}|}{|X|}\right) + \sum_{v\in Values(A)}\left(\frac{|X_{C_A,v}|}{|X|}\log_2\frac{|X_{C_A,v}|}{|X_v|} + \frac{|X_{C_B,v}|}{|X|}\log_2\frac{|X_{C_B,v}|}{|X_v|}\right)$$

X_{C_A} and X_{C_B} denote sets of documents that are included in categories C_A and C_B. $X_{C_A,v}$ and $X_{C_B,v}$denote taking feature value v and its belonging to categories C_A and C_B respectively. Finally, the best k words according to this metric are chosen as features.

4. Evaluation

4.1 Experimental setting

This section evaluates the performance of our algorithms by measuring its ability to reproduce manual category assignments on a data set.

We will now describe the data sets and the method of evaluation. The data set is a set of papers from IJCAI'01 proceedings. We used 188 papers that had extracted titles, authors, and abstracts from PDF files as data. These papers had been manually indexed by category (14 categories). Each category corresponded to a section of IJCAI'01 Proceedings and selection was done as follows: *Knowledge Representation and Reasoning, Search, Satisfiability, and Constraint Satisfaction Problems, Cognitive Modeling, Planning, Diagnosis, Logic Programming and Theorem Proving, Uncertainty and Probabilistic Reasoning, Neural Networks and Genetic Algorithms, Machine Learning and Data Mining, Case-based Reasoning, Multi-Agent System, Natural Language Processing and Information Retrieval, Robotics and Perception, Web Applications*.

Our method of feature selection, called ``Binary Category", and another using IG were used over this data set. The method of comparison used the IG metric assessed over the set of all words encountered in all texts, and then the best *k* were chosen words according to that metric. We called this ``Multivalued Category." After the best features were chosen with Multivalued Category and Binary Category. We estimated the accuracy of classification by classifier using kNN and SVM in each case of *k*. SVM training is carried out with the TinySVM [5]. To handle the n-category classification problem, we applied one-versus-rest approach to TinySVM classifier tool.

To estimate accuracy for selected features, we used an n-fold cross-validation. The data set is randomly divided into n sets with approximately equal size. For each ``fold", the classifier is trained using all but one of the n groups and then tested on the unseen group. This procedure is repeated for each of the n groups. The cross-validation score is the average performance across each of the n training runs. We used a 10-fold cross-validation for our experiments.

4.2 Performance measures

N-best accuracy was used for evaluation. We considered two kinds of criteria for accuracy, ``N=1" and ``N=3".

- ``N=1" meant that the most suitable category predicted by kNN and SVM corresponded to the original target document category, then a correct prediction was considered.
- ``N=3" meant that at least one of three higher suitable categories that were predicted by kNN and SVM corresponded to the original target document category, a correct prediction was considered.

4.3 Experimental Results

Figure 5, Figure 6, Figure 7, and Figure 8 have the results obtained through the different feature selection methods we tested. The results using the kNN classifier are presented in Figure 5 and Figure 6. The other results, Figure 7 and Figure 8, are used the SVM classifier. The horizontal axis is the number of features, and the vertical axis is the accuracy score(\%) for ``N=1" and ``N=3".

Additionally, we experimented the classification of kNN and SVM using an unbounded number of features. The results of the experiments were that Accuracy scores of kNN classification were ``N=1" : $36.7\%$ and ``N=3" : 61.2%, those of SVM classification were ``N=1" : $35.6\%$ and ``N=3" : 61.0%. The accuracy of using an unbounded number of features was lower than that of feature selections. For the result given above, feature selection was proven helpful in improving classification. Furthermore, almost every result of accuracy scores was Binary Category method > Multivalued Category method. In almost all cases, ``N=1" results of Figure 5 and Figure 7, revealed a higher accuracy for the Binary Category method than for the Multivalued Category method. Moreover, the Binary Category method at ``N=3", Figure 6 and Figure 8, was much more accurate than the Multivalued Category method with a fewer number of features. This helped to reduce the impact of noise and irrelevant data, and therefore our feature selection method could reduce the computation costs of classifying new documents without reducing accuracy.

For comparison of kNN (Figure 5, Figure 6) and SVM (Figure 7, Figure 8), their accuracy performance is approximate equivalent. This result was in agreement with [12].

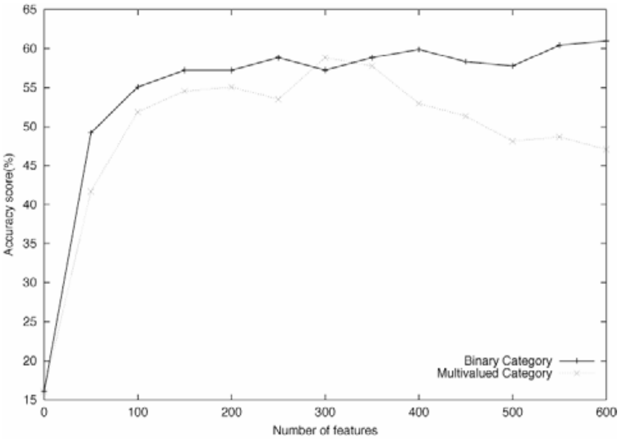


Fig. 5. kNN : accuracy of ``N=1"

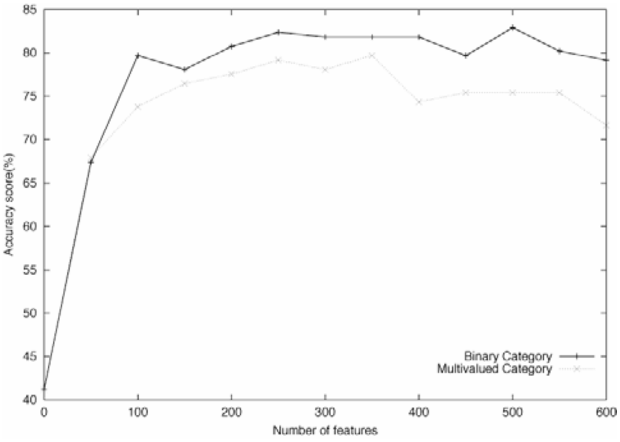


Fig. 6. kNN : accuracy of ``N=3"

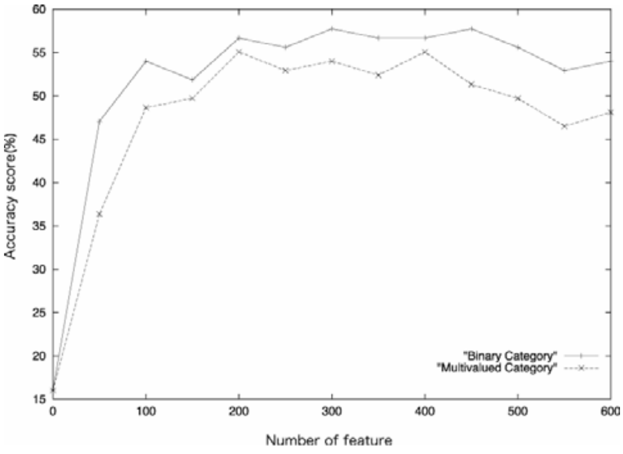


Fig. 7. SVM : accuracy of ``N=1"

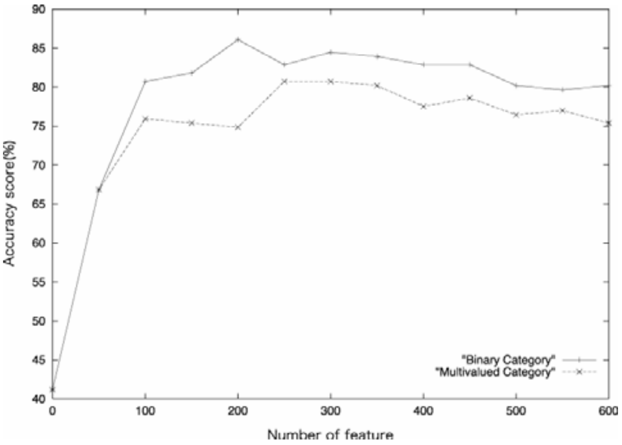


Fig. 8. SVM : accuracy of ``N=3"

5. Discussion

The Papits classifier uses kNN instead of SVM. From accuracy performance point of view, there is not so much difference between kNN and SVM, base on the result of the experiments. Furthermore, [12] describes kNN and SVM have an almost equivalent performance. If SVM is applied to the Papits classifier, Papits has to generate a new classifier whenever users input new paper information or correct that information. Hence Papits uses the kNN algorithm.

In the early stages of Papits running, there may also not be enough registered papers as training data to identify the most suitable words to classify the papers into the multivalued category in Papits. The proposed method in this paper solves this problem. Our method transforms the multivalued category into a binary category. Because of increasing the number of data in one category, our method makes it relatively easy to identify the characteristic words to classify category. Though, the system manager has to input some amount of paper information.

6. Related Works

Feature selection is helpful in reducing noise in document representation, improving both classification and computational efficiency. Therefore, several methods of feature selection have been reported [4][11][13]. Yang [13] reported a comparative study of feature selection methods in statistical learning of text categorization. This paper proposed methods that selected any feature with an IG that was greater than some threshold. Soucy [11] presented methods that combined IG and the cooccurrence of words. This method selects a set of features according to an IG criterion, and refines them based on the cooccurrence with a predetermined subset of highly ranked features. This method evaluates a task of binary classification. Text classification in Papits needs to classify documents to be classified into the multivalued category. Hence it is hard to collect enough training data. Our method considers the case that Papits stores a few training data, transforms the multivalued category into a binary category to easily identify the characteristic words. John [4] proposed feature selection in the wrapper model. This method finds all strongly suitable features and a useful subset of the weakly relevant features that yields good performance. The processing cost to identify weakly relevant features was very expensive, because the wrapper model repeats evaluation with respect to every subset of features. Our method considered subsets of categories. Subsets of categories are much smaller than that of features.

7. Conclusions and Future Work

In this paper, we introduced an approach and a structure to implement automatic classification in Papits. This structure gradually increased the accuracy by using feedback from users. In this system, papers classified by the classifier were not used as training data, since these cannot guarantee a perfectly correct prediction. An unclassified paper is classified by a classifier that only uses manually classified papers in the document DB as training data.

The main problem for the automatic text classification is to identify what words are most suitable to classify documents in predefined classes. Automatic classification in Papits needs to classify documents into the multivalued category, since research is organized by field. To solve this problem, we proposed a feature selection method for text classification in Papits. It transforms the multivalued category into a binary category and was helpful in reducing noise in document representation and improving classification and computational efficiency, because it increased the amount of data in one category, and selected features using IG. We experimentally confirmed its efficacy.

One direction for future study is to develop a means of determining parameters that are suited to the task required, such as the number of features and the number of combinations of categories.

References

[1] C. J. C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery, 2(2),pp.121-167 1998

[2] N. Fujimaki, T. Ozono, and T. Shintani, Flexible Query Modifier for Research Support System Papits., Proceedings of the IASTED International Conference on Artificial and Computational Intelligence(ACI2002), pp.142-147, 2002.

[3] T. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Proceedings of the European Conference on Machine Learning, 1998.

[4] G. H. John, R. Kohavi, K. Pfleger, Irrelevant Features and the Subset Selection Problem, Proceedings of the Eleventh International Conference on Machine Learning, pp.121-129, 1994.

[5] T. Kudo, TinySVM: Support Vector Machines, http://cl-aist-nara.ac.jp/~taku-ku/software/TinySVM ,2001

[6] D. Lewis and M. Ringuette, A comparison of two learning algorithms for text categorization, Third Annual Symposium on Document Analysis and Information Retrieval, pp 81-93, 1994.

[7] K. Nigam, J. Lafferty and A. McCallum, Using Maximum Entropy for Text Classification, IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999.

[8] T. Ozono, S. Goto, N. Fujimaki, and T. Shintani, P2P based Knowledge Source Discovery on Research Support System Papits, The First International Joint Conference on Autonomous Agents & Multiagent Systems(AAMAS 2002), 2002.

[9] J. R. Quinlan, Induction of decision trees, Machine Learning, 1 (1) pp 81-106, 1986.

[10] M. Sahami, S. Dumais, D. Heckerman and E. Horvitz, A Bayesian approach to filtering junk e-mail, AAAI/ICML Workshop on Learning for Text Categorization, 1998.

[11] P. Soucy and G. W. Mineau, A Simple Feature Selection Method for Text Classification, Proceedings of International joint Conference on Artificial Intelligence(IJCAI'01), pp. 897-902, 2001.

[12] Y. Yang and X. Liu, A re-examination of text categorization methods, 22nd Annual International SIGIR, pp.42-49, 1999.

[13] Y. Yang and J. O. Perdersen, A Comparative Study on Feature Selection in Text Categorization., Proceedings of the Fourteenth International Conference on Machine Learning(ICML'97), 1997.

Tadachika Ozono received his bachelor degree in engineering from Nagoya Institute of Technology, his master degree in engineering from Nagoya Institute of Technology, and his Ph.D in engineering from Nagoya Institute of Technology. He is a research associate of the Graduate School of Computer Science and Engineering at Nagoya Institute of Technology. His research topic is a web intelligence using multiagent and machine learning technologies.