International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:3, 2009

Impovement of a Label Extraction Method for a Risk Search System

Shigeaki Sakurai, and Ryohei Orihara

Abstract**—**This paper proposes an improvement method of classification efficiency in a classification model. The model is used in a risk search system and extracts specific labels from articles posted at bulletin board sites. The system can analyze the important discussions composed of the articles. The improvement method introduces ensemble learning methods that use multiple classification models. Also, it introduces expressions related to the specific labels into generation of word vectors. The paper applies the improvement method to articles collected from three bulletin board sites selected by users and verifies the effectiveness of the improvement method.

Keywords**—**Text mining, Risk search system, Corporate reputation, Bulletin board site, Ensemble learning

I. Introduction

Owing to the progress of internet environments and computer environments, many texts are uploaded to the Web. There is a need to analyze the texts, because new knowledge can be buried in the texts. Hu and Liu [6] proposed a method that analyzes customer reviews on the Web. The method mines product features, identifies opinion sentences, and summarizes the results. Morinaga et al. [9] proposed a method that analyzes product reputations. The method extracts opinions by using syntactic and linguistic rules. The method attaches the positive/negative determination, the product name, and the degree of the confidence to the opinions. Also, Kobayashi et al. [7] proposed a method that extracts attribute-value pairs. The method judges whether the pairs express an opinion of the author. In addition, Esuli and Sebastiani [2] proposed SENTIWORDNET for opinion mining. It is a lexical resource in which WordNet synset [8] is associated with three numerical scores. The scores show how objective, positive, and negative the terms contained in the synset are.

On the other hand, we have been studying a method of analyzing corporate reputations on bulletin board sites. Sakurai and Orihara [10] proposed a method that extracts labels from articles posted at bulletin board sites, decides important threads based on extracted labels, and extracts characteristic expressions of the threads. Also, Sakurai and Orihara [11] proposed a method that revises both the decision as to which threads are important and the extraction of characteristic expressions. Sakurai and Orihara [10] [11] and Sakurai [12] verified its effectiveness. This paper proposes the method, aiming at the improvement of label extraction in order to decrease the workload for checking threads, and verifies the effectiveness of the method through numerical experiments.

Shigeaki Sakurai is with the System Engineering Laboratory, Corporate Research & Development Center, Toshiba Corporation, Kawasaki, e-mail: shigeaki.sakurai@toshiba.co.jp. Ryohei Orihara is with the HumanCentric Laboratory, Corporate Research & Development Center, Toshiba Corporation, Kawasaki, e-mail: ryohei.orihara@toshiba.co.jp.

II. Risk Search System

A. Target Sites

The risk search system deals with bulletin board sites selected by the users. Each site includes one or more threads and each thread is composed of thread information and one or more articles. Here, the thread information is composed of a URL and the thread title. Also, each article is composed of attributes, time, and text. The attributes show features of the article such as the author name and the article title, the time means the time stamp indicating when the article was uploaded, and the text shows the body of the article. Figure 1 shows an example of a thread.

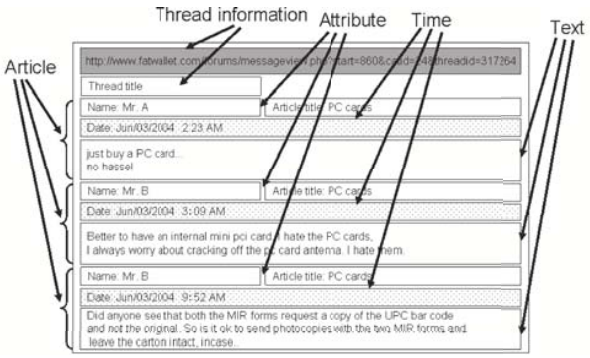


Fig. 1. An example of a thread at a target site

B. System Outline

As shown in Figure 2, the proposed risk search system is composed of three components: the thread crawler, the thread analyzer, and the analysis result viewer. The system downloads threads from target sites selected by users, extracts articles from the threads, and outputs the articles to the article database (DB) with the specific format. Also, the system analyzes the articles by referring to the analysis knowledge DB and outputs analysis results to the analysis result DB. Lastly, the system shows the analysis results to the users. The users can judge how the thread should be processed by referring to the results and the other information such as inquiry information and design information of products. In the following subsection, this paper briefly explains each component.

C. Thread Crawler

The thread crawler extracts threads from target sites and extracts articles included in the threads. In the risk search

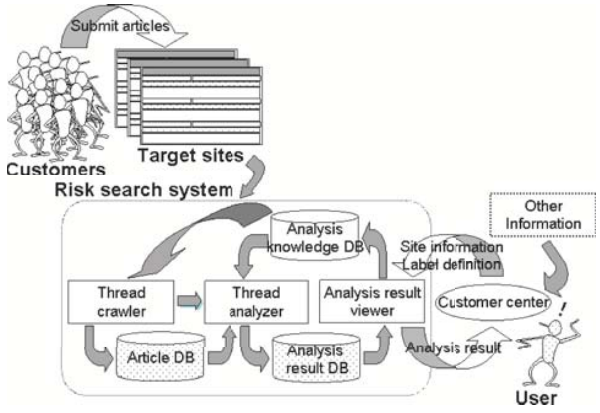


Fig. 2. An outline of the risk search system

system, the number of the target sites is at most 10 because the number of important bulletin board sites is limited. Also, the component extracts attributes, time, and text from each article. Then, the component uses HTML wrapper functions to extract them. Each wrapper function is manually generated for each bulletin board site. The workload for generating the wrapper function is not big, because the number of the target sites is not big and their design is not altered frequently. In addition, we can generate the wrapper functions without updating basic programs of the risk search system by using the generation method based on XQuery.

D. Thread Analyzer

The thread analyzer analyzes articles in the article DB and outputs analysis results to the analysis result DB by referring to the analysis knowledge DB. The component is composed of three processes and the analysis knowledge DB is composed of five kinds of background knowledge as shown in Figure 4.

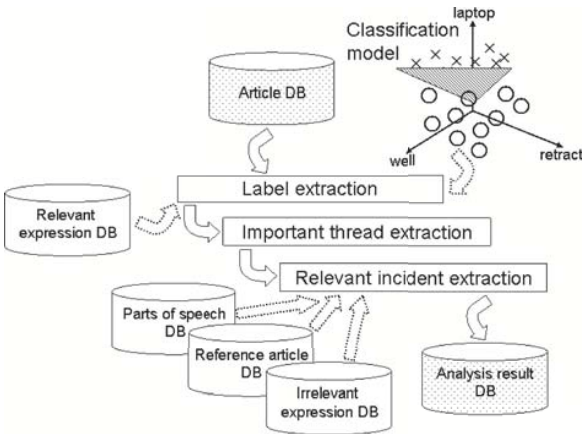


Fig. 3. An outline of the thread analyzer

At first, the label extraction extracts labels from texts of each article. In order to avoid overlooking important threads, the process uses two extraction methods: a classification model-based method and a relevant expression-based method. The classification model-based method applies the morphological analysis to texts, and generates word vectors. The word vectors indicate whether selected words are included in the texts or not. The method applies the word vectors to classification models of specific labels and decides whether the labels are assigned to the texts. The models are acquired inductively by using an inductive learning method: SVM (Support Vector Machine) [14]. On the other hand, the relevant expression-based method applies the morphological analysis to both texts included in articles and relevant expressions stored in the relevant expression DB. The relevant expressions are either words or phrases related to specific labels. The method judges whether texts include their word stems accompanied by their parts of speech, and decides whether the corresponding labels are assigned to the texts. Lastly, the process integrates labels extracted by these two methods.

Next, the important thread extraction decides important threads and decides their order based on the frequency of predefined labels. In the risk search system, the predefined labels are a specific company label and a complaint label. The process evaluates which company is mainly discussed in a target thread. The process regards the target thread as a candidate important thread when the evaluated company corresponds to the predefined company label. Also, the process evaluates the number of complaint labels included in the candidate important thread. The process regards the candidate important thread as an important thread when the number is larger than or equal to a predefined threshold. The process arranges extracted important threads in the descending order of the number of complaint labels.

Lastly, the relevant incident extraction extracts relevant incidents from extracted important threads. The process extracts rows of words from articles included in an important thread by referring to the parts of speech DB. The DB stores rows of parts of speech decided on the basis of linguistic background knowledge. “adjective + noun” and “adverb + verb” are examples of the rows. The process compares the numbers of the rows of words in the important thread with the ones in the reference articles and evaluates whether the rows are candidate relevant incidents or not. The reference articles are a set of articles collected from many bulletin board sites and are stored in the reference article DB. The process extracts rows that are frequent in the important thread, but are not frequent in the reference articles, as candidate relevant incidents. Also, the process gets rid of candidate relevant incidents included in the irrelevant expression DB. The DB stores the expressions that are clearly not of interest to the users. The process extracts remaining incidents as relevant incidents.

E. Analysis Result Viewer

The analysis result viewer shows the analysis results to the users by using three views: the extracted label view, the article time series view, and the relevant incident view. The extracted label view shows whether articles include complaint labels or specific company labels. The view collects articles included in the same thread and arranges the articles in the ascending order of the time. The article time series view shows the

change of both the number of articles included in an important thread and the number of complaint articles included in it. The view collects articles included in the same year and the same month, and arranges the numbers in the ascending order of the years and the months. The relevant incident view shows relevant incidents included in an important thread. The view arranges the relevant incidents in the descending order of the evaluation values. The view also shows a product label of the important thread, total number of articles included in it, and total number of complaint articles included in it. The product label corresponds to a product which is mainly discussed in the important thread. The view arranges important threads in the descending order of the number of complaint articles. Figure 4 shows an example of the analysis result viewer. In this graph, the upper graph is the extracted label view, the lower-left graph is the article time series view, and the lower-right graph is the relevant incident view.

These views enable the users to grasp important threads and the outline of the contents described in them. Also, the users can check articles including a specific company label and a complaint label if the users try to grasp the contents in detail.

III. Improvement of Label Extraction

In the risk search system, labels are important features of threads and have a large impact on the analysis efficiency of the system. Even if our previous methods attained a level such that the system was suitable for use in daily operations, the extraction of more valid levels leads to the decrease of the workload for checking the threads. Thus, this section proposes the improvement method of the label extraction based on the classification model.

A. Introduction of Ensemble Learning

In machine learning research, the effect of ensemble learning techniques has been verified. The techniques acquire multiple classification models from a training example set and infer a class for a target evaluation sample by using the acquired classification models. The techniques give higher classification efficiency than the case of single classification model does. Bagging [1] and boosting [3] are typical techniques. We can anticipate that the introduction of the techniques leads to the improvement of the label extraction in the risk search system.

Bagging generates a subset of a training example set based on a sampling method, where the sampling method can select the same training example repeatedly. Bagging acquires a classification model from the subset. Also, bagging repeats both the sampling and the acquisition until the number of the acquired classification models reaches the predefined threshold. In the inference phase, bagging applies a target evaluation sample to each classification model and assigns a majority class to the sample.

On the other hand, boosting aims at acquiring the classification model for training examples that are classified into wrong classes by the present classification models. We note AdaBoost [4], one of the best-known boosting algorithms. AdaBoost assigns the same weights to training examples and, firstly, acquires the classification model from the training examples assigned the weights. Also, AdaBoost calculates α_1 of the classification model based on Formula (1), where e_1 is the weighted error rate of the classification model.

$$\alpha_1 = \frac{1}{2}\log_e\left(\frac{1-e_1}{e_1}\right) \tag{1}$$

If a training example is misclassified by the classification model, AdaBoost updates its weight by multiplying $e^{-\alpha_1}$. Otherwise, AdaBoost updates its weight by multiplying e^{α_1}. Next, AdaBoost acquires the second classification model from the training examples updated by the weights and calculates α_2. AdaBoost repeats both the acquisition and the update, and acquires the classification models whose number reaches the predefined threshold. In the inference phase, AdaBoost applies a target evaluation sample to each classification model and judges a class corresponding to the classification model. It also adds up values of α_t, (t=1, 2, $\cdots$) for each class and assigns the class with the maximum value to the sample.

We need to design concrete sampling methods and learning methods based on the weighted training examples in order to incorporate the techniques into the label extraction in the risk search system. This paper evaluates 4 sampling methods: the equality division method of negative examples, the random division method of negative examples, the random division method of all examples, and the random division method of positive examples and negative examples. In these methods, we assume that the number of the negative examples (n) is larger than the number of the positive examples (p). The positive examples are examples that include a specific label and the negative examples are examples that do not include the specific one. In the risk search system, only a few parts of articles include the specific label. In the following, each method is explained.

L1. The equality division method of negative examples:

(a) Let the number of the classification models (c) be the number that is round $r = \frac{n}{p}$ to the integer.

(b) Equally divide negative examples into c subsets.

(c) Integrate all positive examples and one of the subsets, and generate c training example subsets.

Figure 5 shows an outline of the equality division method of negative examples. In this figure, grey cylinders show negative example sets and white cylinders show positive example sets.

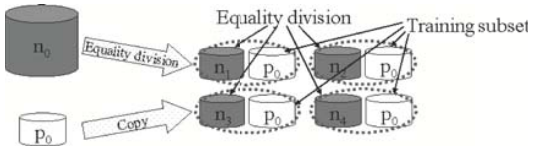


Fig. 5. An outline of the equality division method of negative examples

L2. The random division method of negative examples:

(a) Let the number of the classification models (c) be the number that is round $r = \frac{n}{p}$ to the integer.

(b) Calculate $\frac{1}{r}$.

(c) Repeat the following steps (d) $\sim$ (f) c times and generate c training example subsets.

(d) Initialize a training example subset.

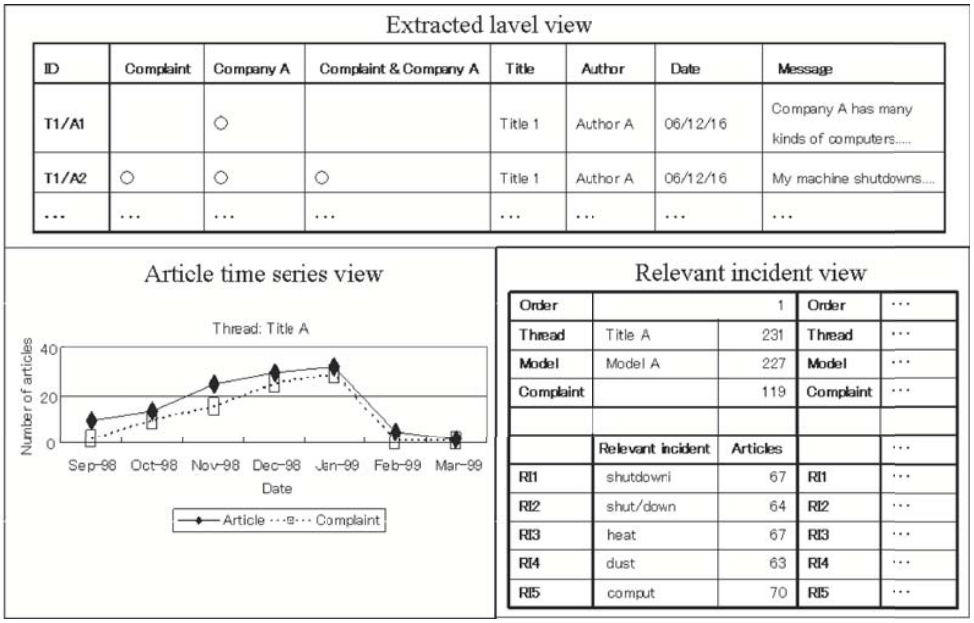


Fig. 4. Analysis result viewer

(e) For each negative example, decide the random value $\in [0, 1]$ and compare the value with $\frac{1}{r}$. If the value is larger than or equal to $\frac{1}{r}$, add the negative example to the training example subset. Otherwise, ignore the negative example.

(f) Add all positive examples to the training example subset.

Figure 6 shows an outline of the random division method of negative examples.

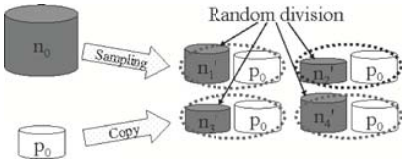


Fig. 6. An outline of the random division method of negative examples

L3. The random division method of all examples:

(a) Set the number of the classification models (c) and the sampling rate (s).

(b) Decide the number of training examples included in each training example subset (x) based on s.

(c) Repeat the following steps (d) c times and generate c training example subsets.

(d) Randomly select x training examples from all training examples, where the same training examples can be selected repeatedly.

Figure 7 shows an outline of the random division method of all examples. In this figure, checked cylinders show sets composed of both negative examples and positive examples.

L4. The random division method of positive examples and negative examples:

(a) Set the number of the classification models (c) and the sampling rate (s).

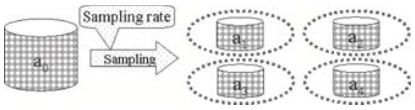


Fig. 7. An outline of the random division method of all examples

(b) Decide the number of training examples included in each training example subset (x) based on s.

(c) Repeat the following steps (d) c times and generate c training example subsets.

(d) Repeat the following steps (e) $\sim$ (f) x times.

(e) Randomly decide positive or negative.

(f) Randomly select a training example from training examples corresponding to the selected class, where the same training examples can be selected repeatedly.

Figure 8 shows an outline of the random division method of positive examples and negative examples.

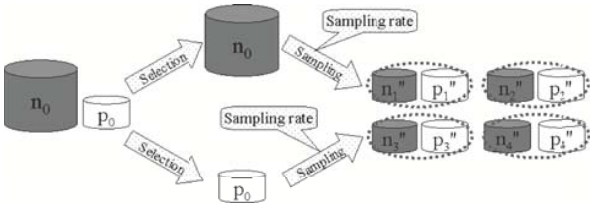


Fig. 8. An outline of the random division method of positive examples and negative examples

In these methods, L3 is the method that most naturally expresses the concept of bagging. The other methods are not always normal sampling. However, in the case of the collected training example set, only a few parts of the set are positive examples. Inductive learning algorithms tend to acquire imbal-

anced classification models. That is, the classification models tend to judge that classes of target evaluation samples are negative. We anticipate the methods can relax the imbalance.

Next, we note the case of boosting. The label extraction uses classification models acquired by SVM. SVM cannot deal directly with the weights of the training examples. In this paper, we try to get the effect of the weights by the weighted sampling. The sampling is explained in the following.

L5. The weighted sampling:

(a) Set the number of the classification models (c) and the sampling rate (s).

(b) Set s to weights of training examples.

(c) Repeat the following steps (d) $\sim$ (h) c times and generate c training example subsets.

(d) Initialize a training example subset.

(e) For each training example, repeat the following steps (f) $\sim$ (g).

(f) Extract the integer part of the weight (y) and the decimal part of the weight (z). If y is larger than or equal to 1, add the training example to the subset y times.

(g) Decide the random value $\in [0, 1]$ and compare the value with z. If z is smaller than or equal to the value, add the training example to the subset again.

(h) Update the weights based on the classification model acquired from the training example subset.

Figure 9 shows an outline of the weighted sampling.

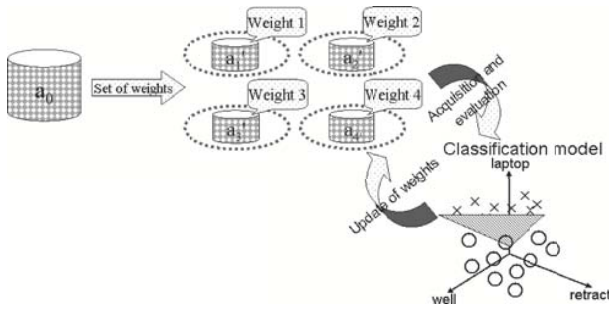


Fig. 9. An outline of the weighted sampling

B. Introduction of Attributes Related to Relevant Expressions

The previous generation of a word vector selects words whose tf-idf (term frequency - inverse document frequency) values [13] are larger than or equal to predefined threshold. Also, the generation ignores the difference of parts of speech, because the articles include many grammatical errors and assigned parts of speech are not always valid. In the case of extraction of a specific label, small tf-idf values may be given to important words, because the tf-idf values are calculated based on the collected articles. The generation may overlook important words for the extraction of the label. In addition, the generation cannot deal with a word set as an attribute.

On the other hand, the risk search system has the relevant expression DB in order to extract specific labels. The DB stores words and word sets related to the specific labels as relevant expressions. If articles include the relevant expressions, the articles are positive examples with high probability. We can anticipate that the attributes corresponding to the relevant expressions lead to the improvement of the classification model. Thus, this paper introduces the attributes based on the relevant expressions. We evaluate the following three methods. In D2 and D3, we introduce the attribute based on all relevant expressions of a specific label. This is because the numbers of articles including respective relevant expressions of the specific label may be much smaller than the numbers of articles including word stems selected as attributes.

D1. The introduction method of attributes corresponding to each relevant expression: The method regards a relevant expression of a specific label as an attribute. If an article includes the relevant expression, the method sets 1 as the attribute value. Otherwise, the method sets 0 as the attribute value. The method generates new attributes whose number is the number of the relevant expressions (m).

Table I shows additional attributes generated by the introduction method of attributes corresponding to each relevant expression. In this table, "R. E." is an abbreviation of a relevant expression.

TABLE I
AN EXAMPLE OF THE INTRODUCTION METHOD OF ATTRIBUTES CORRESPONDING TO EACH RELEVANT EXPRESSION

Attribute	R. E. 1	R. E. 2	$\cdots$	R. E. m
Attribute value	1	0	$\cdots$	1

D2. The introduction method of an attribute corresponding to all relevant expressions: The method regards all relevant expressions of a specific label as an attribute. If an article does not include the relevant expressions at all, the method sets 0 as the attribute value. Otherwise, the method sets 1 as the attribute value. The method generates a new attribute.

Table II shows an additional attribute generated by the introduction method of an attribute corresponding to all relevant expressions.

TABLE II
AN EXAMPLE OF THE INTRODUCTION METHOD OF AN ATTRIBUTE CORRESPONDING TO ALL RELEVANT EXPRESSIONS

Attribute	{ R. E. 1, R. E. 2, $\cdots$, R. E. m}
Attribute value	1

D3. The mixed introduction method: The method uses both attributes based on D1 and an attribute based on D2. That is, the method generates new attributes whose number is m+1.

Table III shows additional attributes generated by the mixed introduction method.

TABLE III
AN EXAMPLE OF THE MIXED INTRODUCTION METHOD

Attribute	R. E. 1	$\cdots$	R. E. m	{ R. E. 1, $\cdots$, R. E. m}
Attribute value	1	$\cdots$	0	1

IV. NUMERICAL EXPERIMENT

In this section, we present a numerical experiment to verify the effect of the ensemble learning and the relevant expressions.

A. Experimental Data

In this numerical experiment, we use 10,002 articles collected from three bulletin board sites: www.hardwareanalysis.com, www.fatwallet.com, and groups.yahoo.com. In order to assign a complaint label to the article, we evaluate the article by reading it. The evaluation leads to 2,298 articles assigned the label and 7,704 articles not assigned the label. We perform the numerical experiment based on 2,298 positive examples and 7,704 negative examples

B. Experimental Method

We compare the previous method (L0D0) with the proposed methods. Here, L0 indicates the method does not use the ensemble learning techniques and D0 indicates the method does not deal with relevant expressions. We perform numerical experiments based on 10-fold cross-validations for the combination of L0 $\sim$ L5 and D0 $\sim$ D3. The SVM software used is libsvm [5]. We select a linear kernel and set default values to its parameters, because the kernel gives comparatively high classification efficiency without adjusting the parameters. Also, we set 0.0001 as a threshold of tf-idf values for a word vector according to previous experimental results. The threshold extracts 3,524 word stems. That is, in the case of D0, the dimension of the word vector is 3,524. In addition, we use 213 relevant expressions of the complaint label stored in the relevant expression DB of the present risk search system.

On the other hand, we evaluate experimental results based on three evaluation criteria: recall, precision, and accuracy. The criteria are defined by Formula (2), Formula (3), and Formula (4).

$$\text{recall} = \frac{p_t}{p_t + n_f} \tag{2}$$

$$\text{precision} = \frac{p_t}{p_t + p_f} \tag{3}$$

$$\text{accuracy} = \frac{p_t + n_t}{p_t + p_f + n_t + n_f} \tag{4}$$

Here, p_t is the number of positive examples whose classes are evaluated as positive by the classification model. p_f is the number of positive examples whose classes are evaluated as negative by it. n_t is the number of negative examples whose classes are evaluated as negative by it. n_f is the number of negative examples whose classes are evaluated as positive by it.

In the case of L3, L4, and L5, we use 3, 5, 7, 9, 11, and 13 as the numbers of the classification models, and we use 0.4, 0.6, 0.8, and 1.0 as the sampling rates. But, in the case that the numbers of the classification models are 11 and 13, we use only 0.8 and 1.0 as the sampling rates. This is because small sampling rates do not give good classification efficiency in other experiments.

C. Experimental Results

Figure 10 shows the parts of the experimental results. The results are average values of 10 experiments based on the 10-fold cross-validations. In each figure, the horizontal line shows the method and the vertical line shows the classification efficiency. Also, M1 $\sim$ M13 show the numbers of the classification models and O0.4 $\sim$ O1.0 show the sampling rates. In the case of L0, the number and the rate are M1 and O1.0, respectively. In the case of L1 and L2, the number and the rate are M3 and O0.5, respectively

D. Discussions

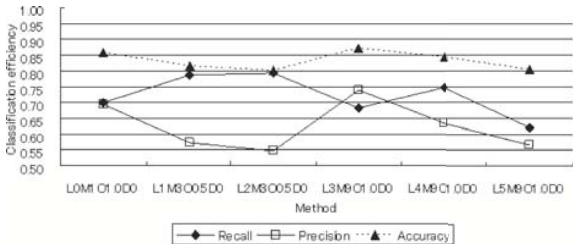
Effect of ensemble learning techniques: We note Figure 10(a) $\sim$ Figure 10(d). L1, L2, and L4 improve the recall, but they aggravate the precision. The present risk search system still realizes high recall. The improvement of the precision is more important than that of the recall. These methods are ineffective for improving the classification efficiency of the system. Also, L5 aggravates all evaluation criteria and cannot get the effect we anticipated. The over-fitting in the boosting may cause the aggravation or the usage method of the weights may be inappropriate. In future work, we intend to investigate the cause in greater detail. On the other hand, L3 improves precision 4% and keeps recall close to the recall in the case of L0. We think that the multiple classification model based on the bagging compensated for the error of the individual classification model. The method is efficient for the present risk search system.

We introduced L1, L2, and L4 in order to adjust the imbalance of negative examples. However, the results show that the adjustment is over-fitting for positive examples. The methods based on multiple classification models have the effect of the adjustment and excessive adjustment may occur.

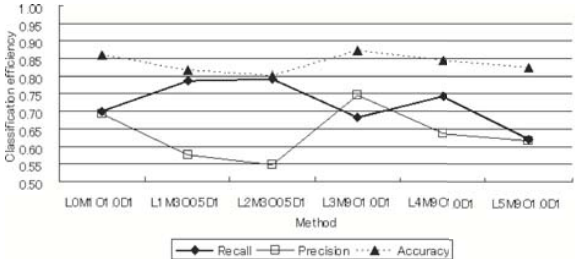
Effect of parameters in the ensemble learning: We note the difference of the parameters in the case of L3. At first, we note Figure 10(e). The results show that the recall improves but the precision deteriorates as the sampling rates increase. In the case of high sampling rates, various positive examples are included in the training example subset and it is easy for the classification models to judge that the target evaluation sample is positive. The classification models can give high recall. In this paper, we aim at the improvement of the precision and keeping high recall. The low sampling rates violate the latter condition. Therefore, we think the appropriate sampling rate is 1.0.

Next, we note Figure 10(f). The results show that the classification efficiency improves as the number of classification models increases. However, the improvement is small in the case that the number is 13. On the other hand, the number of classification models has a big influence on calculation time. That is, the bigger the number is, the longer the calculation time becomes. In the present risk search system, the amount of the calculation for analyzing articles is not big. We can use more classification models. Therefore, we think the appropriate number of the classification models is 9 or 11.

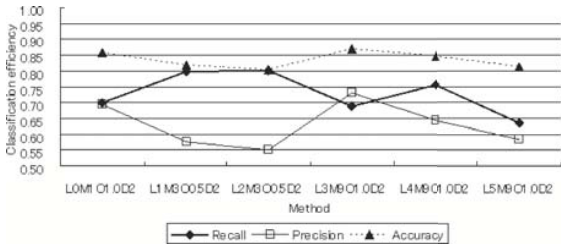
Effect of relevant expressions: We note Figure 10(g) and Figure 10(h). The results show the usage of relevant expressions improves the classification efficiency, but the degree of the improvement is very small. The impact of the usage of relevant expression is less than we anticipated. In the experiments, the word vector is composed of 3,524 word



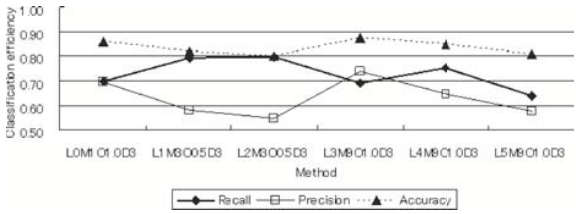
(a) Change of the classification efficiency due to learning methods in the case of D0



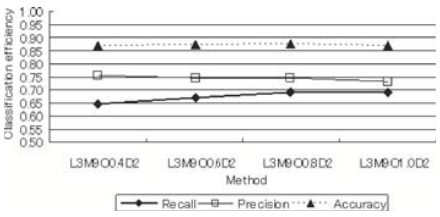
(b) Change of the classification efficiency due to learning methods in the case of D1



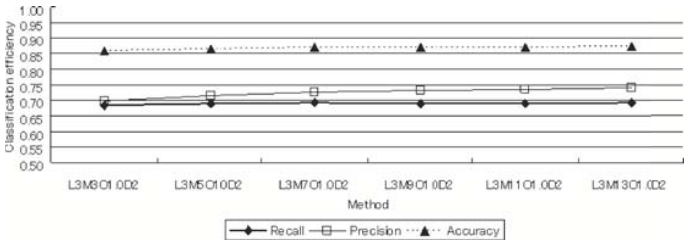
(c) Change of the classification efficiency due to learning methods in the case of D2



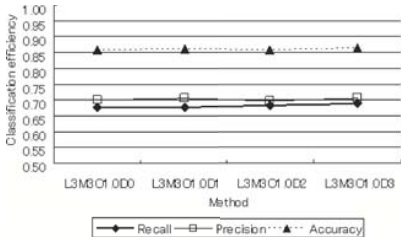
(d) Change of the classification efficiency due to learning methods in the case of D3



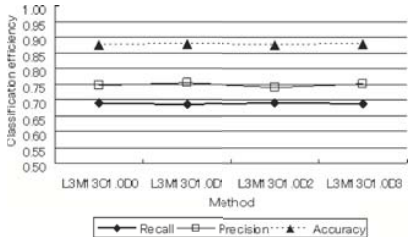
(e) Change of the classification efficiency due to sampling rates in the case of L3



(f) Change of the classification efficiency due to numbers of classification models in the case of L3



(g) Change of the classification efficiency due to usage of relevant expressions in the case of M3



(h) Change of the classification efficiency due to usage of relevant expressions in the case of M13

Fig. 10. Experimental results

stems. It may have still expressed the features of articles sufficiently. In addition, in the present risk search system, the relevant expression DB mainly stores simple relevant expressions. Many parts of word stems corresponding to the relevant expressions may be included in the word vector.

In light of these discussions, we believe the revised classification models can extract labels of greater validity than the present ones do. Also, we believe the models can decrease the workload for checking the threads.

V. Summary and Future Work

This paper introduced the ensemble learning techniques and the relevant expressions in order to improve the label extraction based on the classification models. Also, this paper verified the effectiveness of their introduction by applying them to articles collected from three bulletin board sites. The experimental results indicate that the ensemble learning techniques can greatly improve the classification efficiency, but the relevant expressions cannot always improve it. The introduction of these techniques keeps the recall at a high level and improves precision 4%.

In future work, we intend to use WordNet [8] to refer to relationships between articles in order to more validly analyze the articles. Also, we intend to analyze information on the Web other than company reputation information.

REFERENCES

[1] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996.
[2] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," *Proc. 5th Conf. on Language Resources and Evaluation*, 2006, Genoa, Italy, pp. 417-422.
[3] Y. Freund, "Boosting a Weak Learning Algorithm by Majority," *Information and Computation*, vol. 121, no. 2, pp. 256-285, 1995.
[4] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.
[5] C. -W. Hsu, C. -C. Chang, and C. -J. Lin, "A Practical Guide to Support Vector Classification," *http://www.csie.ntu.edu.tw/˜cjlin/papers/guide/guide.pdf*, 2008.
[6] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," *Proc. 10th Intl. Conf. on Knowledge Discovery and Data Mining*, 2004, Seattle, Washington, USA, pp. 168-177.
[7] N. Kobayashi, R. Iida, K. Inui, and Y. Matsumoto, "Opinion Extraction Using a Learning-Based Anaphora Resolution Technique," *Proc. 2nd Intl. Joint Conf. on Natural Language Processing*, 2005, Jeju Island, Korea, pp. 175-180.
[8] G. A. Miller, C. Fellbaum, R. Tengi, P. Wakefield, H. Langone, and B. R. Haskell, "WordNet," *http://wordnet.princeton.edu/*, 2006.
[9] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining Product Reputations on the Web," *Proc. 8th Intl. Conf. on Knowledge Discovery and Data Mining*, 2002, Edmonton, Alberta, Canada, pp. 341-349.
[10] S. Sakurai and R. Orihara, "Discovery of Important Threads from Bulletin Board Sites," *Intl. J. of Information Technology and Intelligent Computing*, vol. 1, no. 1, pp. 217-228, 2006.
[11] S. Sakurai and R. Orihara, "Discovery of Important Threads using Thread Analysis Reports," *Proc. 2006 IADIS Intl. Conf. of WWW/Internet 2006*, 2006, Murcia, Spain, vol. 2, pp. 243-248.
[12] S. Sakurai, "A Risk Analysis Method using Textual Data on Bulletin Board Sites," *Proc. 8th Intl. Sympo. on advanced Intelligent Systems*, 2007, Sokcho, Korea, pp. 99-102.
[13] G. Salton and M. J. McGill, "Introduction to Modern Information Retrieval," *McGraw Hill Computer Science Series*, 1983.
[14] V. N. Vapnik, "The Nature of Statistical Learning Theory," *Springer*, 1995.

Shigeaki Sakurai received an MS degree in mathematics and a Ph.D. degree in industrial administration from Tokyo University of Science, Japan, in 1991 and 2001, respectively. He became a Professional Engineer of Japan in the field of information engineering in 2004.

He is a research scientist at the System Engineering Laboratory, Corporate Research & Development Center, Toshiba Corporation. His research interests include data mining, soft computing, and web technology.

Dr. Sakurai is a member of IEICE, SOFT, and JSAI.

Ryohei Orihara received a BS degree, an MS degree, and a Ph.D. degree in engineering from the University of Tsukuba, Japan, in 1986, 1988 and 1999, respectively.

He is the laboratory leader at the HumanCentric Laboratory, Corporate Research & Development Center, Toshiba Corporation. He is also a part-time associate professor at Tokyo Institute of Technology, Japan. His research interests include machine learning, creativity support systems, analogical reasoning, metaphor understanding, data mining, and text mining.

Dr. Orihara is a member of IPSJ, JSAI, and JSSST.