Available online at www.sciencedirect.com
SCIENCE DIRECT®
Applied Soft Computing 6 (2005) 62–71
Applied Soft Computing
www.elsevier.com/locate/asoc

An e-mail analysis method based on text mining techniques

S. Sakurai*, A. Suyama

Corporate Research and Development Center, Toshiba Corporation, 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan

Received 1 October 2003; received in revised form 9 August 2004; accepted 1 October 2004

Abstract

This paper proposes a method employing text mining techniques to analyze e-mails collected at a customer center. The method uses two kinds of domain-dependent knowledge. One is a key concept dictionary manually provided by human experts. The other is a concept relation dictionary automatically acquired by a fuzzy inductive learning algorithm. The method inputs the subject and the body of an e-mail and decides a text class for the e-mail. Also, the method extracts key concepts from e-mails and presents their statistical information. This paper applies the method to three kinds of analysis tasks: a product analysis task, a contents analysis task, and an address analysis task. The results of numerical experiments indicate that acquired concept relation dictionaries correspond to the intuition of operators in the customer center and give highly precise ratios in the classification.

Keywords: Text mining; Fuzzy inductive learning; E-mail; Customer center

1. Introduction

Recently, the decisive importance of delivering the highest possible level of customer satisfaction has been widely recognized. Consequently, customer centers dealing with the requests and complaints of customers are assuming a more important role. At the same time, the number of inquiries made to customer centers via e-mail is rapidly increasing. There are two reasons for this increase: e-mail is increasingly selected as the inquiry medium for companies and it makes it easier for customers to make inquires. It is becoming difficult for customer centers to process inquiries and analyze them. Customer centers need a method that classifies e-mails, facilitates their analysis, and transfers them to the appropriate departments.

An e-mail is composed of date, e-mail address, subject, body of the e-mail, and so on. It is possible for the body to include pictures, sounds, and programs, but the body is mainly composed of textual data. Thus, it is possible to use text mining techniques [1,4,7,15] in order to analyze e-mails [10,17]. One paper [17] proposed a method that uses the number of words, the number of lines, and the frequency of important keywords as characteristic values and identifies the person who wrote the e-mail. Another paper [10]

* Corresponding author. Tel.: +81 44 549 2398; fax: +81 44 520 1308.

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2004.10.007

proposed a method that evaluates each word by using the *age* of its word, deletes unimportant words from a word vector, and classifies e-mails based on the word vector. Here, the age of a word is calculated based on the arrival times of e-mails including the word. These methods are not sufficiently able to analyze e-mails collected in a customer center, because the e-mails are sent from many people and include various expressions. That is, even if two e-mails have the same meaning, they may not include common keywords. Also, even if these methods classify e-mails appropriately, it is difficult to understand the validity of classified results by referring to a simple word vector.

On the other hand, some papers [8,11] uses Support Vector Machine (SVM) [2,16] to classify the textual data and show that SVM is an appropriate classifier. Also, Latent Semantic Indexing (LSI) [3], and Probabilistic Latent Semantic Indexing [5] (PLSI) are used to characterize textual data in the field of natural language processing. These methods may be helpful for the analysis of e-mails. However, it is difficult for users to judge whether a classification model given by SVM is valid or not, because SVM acquires hyperplanes, which discriminate classes of items of the textual data in high dimensional space. These indexing methods generate characteristic vectors by integrating words and phrases included in items of textual data into some values. It is difficult for users to intuitively understand relationships between words and phrases, and the items, because the relationships are buried in the characteristic vectors.

Thus, this paper proposes a method to analyze e-mails based on text mining techniques [7,12]. The method generates training examples from training e-mails and their classes based on a key concept dictionary. The method acquires a concept relation dictionary from the training examples by using a fuzzy inductive learning algorithm. The acquired concept relation dictionary is described in the format of a fuzzy decision tree. Users are able to understand the concept relation dictionary easily. The method also classifies new e-mails to be evaluated by using the acquired dictionaries and presents statistical information related to each class. The paper demonstrates the effectiveness of the method by applying it to three kinds of analysis task: a product analysis task, a contents analysis task, and an address analysis task.

2. A text mining method

2.1. A text mining flow

The text mining method [7,12] classifies textual data into some text classes by using a flow shown in Fig. 1. The method uses both the lexical analysis and two kinds of domain-dependent knowledge dictionaries: a key concept dictionary and a concept relation dictionary. The key concept dictionary is a kind of thesaurus and is composed of three layers: a concept class, a key concept, and an expression. Each concept class shows a set of concepts that have a common feature, each key concept shows a set of expressions that have the same meaning, and each expression shows important words and phrases for a target problem. It is possible for the dictionary to deal with different expressions based on their meaning. On the other hand, the concept relation dictionary is composed of relations, which have a condition part with some key concepts, and a result part with a text class. The relations are described in the format of a fuzzy decision tree. Here, the tree is composed of two kinds of nodes and branches connecting an upper node to a lower node. One kind of node is called a branch node and has an attribute. The other kind of node is called a leaf node and has classes with degrees of certainty. Each branch has a fuzzy class item corresponding to an attribute of an upper node. The fuzzy class item is composed of a key concept and its membership function. In the tree, a concept relation is expressed by a path from the top node (root node) to a leaf node. Each relation describes complicated meaning created by the combination of key concepts. The complicated meaning shows a viewpoint of text analysis and the viewpoint is corresponding to a text class.

First, the method applies the lexical analysis to the item and decomposes the item into words, because a language, such as Japanese, is described without word segments. Each word is checked to ascertain whether the word is registered in a key concept dictionary. If a word is registered in the dictionary, a key concept corresponding to the word is assigned to the item.

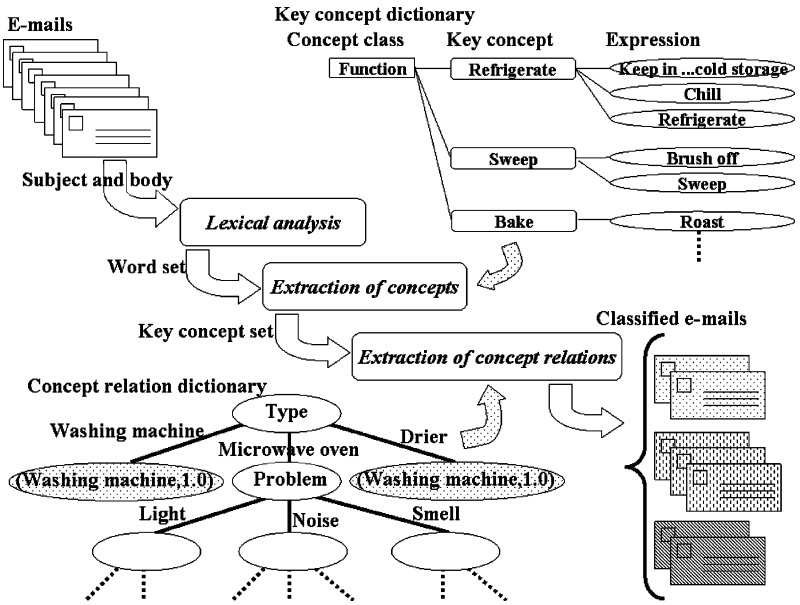

Fig. 1. A text mining flow.

The item is also checked to ascertain whether there is a key concept set described in a concept relation dictionary. If the item has the set, a text class corresponding to the set is assigned to the item. Lastly, the method puts items of textual data that have the same text class into a group.

The method can deal with textual data with grammatical errors, because the method does not uses advanced natural language techniques, such as syntactic analysis. The method also can deal with textual data with polysemy, because the key concept dictionary can store the same expressions in some key concepts and the concept relation dictionary can select a meaning by using a combination of key concepts. Moreover, the method can deal with the ambiguity between key concepts by using the format of a fuzzy decision tree.

2.2. Creation of a key concept dictionary

Although human experts have to manually create a key concept dictionary, they can use a support with a graphical user interface for this task. First, the experts apply the lexical analysis to items of textual data and decompose the items into words. The experts select important words and phrases by referring to frequency of words and tf–idf value [14] and extract them as expressions. Also, the experts create key concepts by gathering the expressions with the same meaning. Moreover, the experts create concept classes by gathering relevant key concepts. The experts check whether the created key concept dictionary extracts or does not extract important expressions by applying the created dictionary to the textual data. If the dictionary does not extract an important expression, the dictionary is revised to extract the expression. The revision of the dictionary is repeated with the aim of extracting all important expressions. The supporting tool provides the environment where the experts perform these processes with a graphical user interface.

2.3. Inductive learning of a concept relation dictionary

A concept relation dictionary is automatically acquired from training examples. Each training example is composed of some key concepts and a text class. The key concepts are extracted from an item of the textual data by using the lexical analysis and a key concept dictionary. Also, users give a viewpoint of analysis to the item by reading it. We regard a concept class as an attribute, a key concept

Table 1
IDTF algorithm

1.	Allocate a training example set to a new node and stack up the node.
2.	Pick out a node from the stack. This algorithm is over, when there is not a node.
3.	Evaluate whether classes with a degree of certainty should be allocated to the node. Let the node be a leaf node allocated to the classes and return to step 2, when it is decided that the classes should be done.
4.	(a) Create such fuzzy sets that represent an attribute for each attribute A_i, where the sets are called fuzzy class items.
	(b) Calculate a *gain_ratio* of each attribute according to both its items and the training example subset allocated to the node.
	(c) Select an attribute with the best evaluation value and allocate the attribute to the node.
	(d) Decompose the training example subset into new subsets according to its items and allocate each subset to a new node.
	(e) Create such branches that connect the original node to each new node and allocate each item to its corresponding branch.
	(f) Stack up the new nodes and return to step 2.

as an attribute value, and a viewpoint as a text class. But, in the case that a key concept included in a concept class is not extracted from the e-mail, a special attribute value of ''nothing'' is allocated to a corresponding attribute. We can acquire a concept relation dictionary from the training examples by using an inductive learning algorithm. The inductive learning algorithm has to deal with a set of attribute values, because key concepts included in the same concept class may be extracted from an item of textual data. We use Induction Decision Tree with Fuzziness (IDTF) [13] to acquire a concept relation dictionary. IDTF is an ID3-like algorithm and is able to deal with discrete values, continuous values, fuzzy values, and set values by defining membership functions for each attribute. IDTF makes a fuzzy decision tree grow by recursively decomposing a training example set into subsets with a selected attribute. That is, IDTF acquires a concept relation dictionary by using the algorithm shown in Table 1. Here, in step 4b, IDTF calculates grades of a training example for fuzzy class items corresponding to an attribute, calculates degrees of certainty of the example for the items by normalizing the grades. IDTF also calculates degrees for all examples included in the subset and calculates a *gain_ratio* [9] by using their total instead of the number of the examples in [9]. In step 4d, IDTF gathers examples whose updated degrees are more than 0 for items corresponding to the selected attribute and generates a new subset corresponding to each item.

On the other hand, the fuzzy class item is composed of a key concept and its membership function defined by Formula (1). Here, v_i is a subset of key concepts included in the ith attribute of an example, L_{ik} is a subset of key concepts included in the ith attribute corresponding to the kth branch node, l_{ikr} is the rth element of L_{ik}, and $|\cdot|$ is an operation that calculates the number of elements included in a set.

$$\begin{aligned} &\text{if } l_{ikr} \in v_i, \quad \text{then} \quad \text{grade}_{ikr} = \frac{1}{|v_i|} + \frac{1-\alpha}{|L_{ik}|} \\ &\text{if } l_{ikr} \notin v_i, \quad \text{then} \quad \text{grade}_{ikr} = \frac{1-\alpha}{|L_{ik}|} \\ &\alpha = \frac{|v_i \cap L_{ik}|}{|v_i|} \end{aligned} \tag{1}$$

This formula has the following meaning. When a key concept included in the ith attribute of an example is equal to one of the key concepts corresponding to the kth branch node, the formula gives a weight $\frac{1}{|v_i|}$ to a lower node connecting to the branch with the key concept. When the key concept in the attribute is not equal to any key concepts corresponding to the branch node, the formula gives an equal weight $\frac{1-\alpha}{|L_{ik}|}$ to all lower nodes connecting to the branch node. Then, we note that L_{ik} is composed of key concepts included in the attribute of examples, which are given to the branch node in the learning phase. That is, α is equal to 1 in the learning phase, because $v_i \cap L_{ik}$ is equal to v_i. On the other hand, $v_i \cap L_{ik}$ is not always equal to v_i in the evaluation phase, because there are key concepts that occur only in the evaluation phase and IDTF does not generate lower nodes corresponding to the key concepts. In this case, it is impossible to evaluate an example in the attribute. So, an equal weight is given to all lower nodes to inspect all possibilities.

For example, four expressions, b_1, c_2, d_1, and b_2, are extracted from an e-mail, and the user gives a text class of $CL1$ to the e-mail. Also, a key concept dictionary is given as input, as shown in Fig. 2. Then,

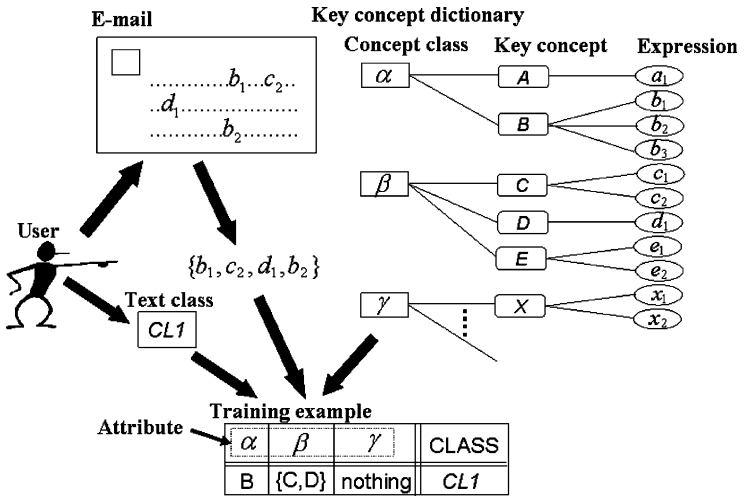

Fig. 2. Generation of a training example.

expressions b_1 and b_2 correspond to a key concept B, an expression c_2 corresponds to a key concept C, and an expression d_2 corresponds to a key concept D. A concept class α has B and a concept class β has C and D. A concept class γ does not have a key concept. So, a training example is generated as output, as shown in Fig. 2.

3. An e-mail analysis system

An e-mail is mainly composed of textual data. Each item of the textual data usually has a lot of grammatical errors, because few people take great care and concentrate when composing e-mails. We are able to analyze e-mails by using the text mining method shown in Section 2. Thus, a new e-mail analysis system based on the method is proposed. The system uses subjects and body of the e-mails. The system is able to divide the e-mails into text classes, which provide a viewpoint for analysis of e-mails. Also, the system extracts key concepts from the e-mails and presents statistical information, such as the distribution of key concepts and the number of e-mails included in each class. Moreover, the system lists e-mails relating to the number. Users are able to read only the e-mails classified into text classes corresponding to their interests. Users are able to understand the trend of the key concepts for all e-mails by referring to the information.

4. Numerical experiments

4.1. Experimental data

The customer center of Toshiba Corp. (our employer) collects e-mails sent by customers, and its own responses, in a database. This experiment uses two kinds of data set in the database. One data set is composed of 466 e-mails. The data set is analyzed with two criteria by operators of the customer center. One criterion is a product criterion that analyzes e-mails with five kinds of text classes, ‘‘Washing machine’’, ‘‘Vacuum cleaner’’, ‘‘Refrigerator’’, ‘‘Microwave oven’’, and ‘‘Others’’. Here, each e-mail assigned to ‘‘Washing machine’’, ‘‘Vacuum cleaner’’, ‘‘Refrigerator’’, and ‘‘Microwave oven’’ has a topic relating to the product. Each e-mail assigned to ‘‘Others’’ has a topic relating to a product other than the four products. In the case that an e-mail included some topics, the operator assigned the e-mail to a text class including the main topic. The other criterion is a contents criterion that analyzes e-mails with five kinds of text class, ‘‘Question’’, ‘‘Request’’, ‘‘Suggestion’’, ‘‘Complaint’’, and ‘‘Others’’. Here, each e-mail assigned to ‘‘Question’’, ‘‘Request’’, ‘‘Suggestion’’, and ‘‘Complaint’’ has a topic relating to four kinds of customer voice. Each e-mail assigned to ‘‘Others’’ has a topic relating to another kind of customer voice, such as ‘‘Thanks’’ or ‘‘Comments’’.

The other data set is composed of 581 e-mails. This data set is analyzed based on the addresses to which e-mails are to be transferred. This criterion identifies e-mails for 12 departments and assigns e-mails, which are not included in these departments to ''Others''.

These e-mails include personal information such as name, address, and telephone number. Care must be exercised in dealing with personal information. Thus, we exclude personal information from the e-mails by replacing the information with special character strings and then use the e-mails without personal information in the experiments. The information is not important for analysis of textual data, because the information is not able to characterize multiple textual data. So, we think the information has little influence on our experiments.

4.2. Key concept dictionaries

We create key concept dictionaries corresponding to each criterion. Table 2 shows the sizes of the key concept dictionaries created by a human expert and the time spent to create the dictionaries. In the table, ''Product'' shows values in the case of the product analysis task, ''Contents'' does so in the contents analysis task, and ''Address'' does so in the address analysis task.

In the case of the product analysis task, the dictionary has concept classes, such as product names, model number of products, and functions of products. In the case of the contents analysis task, the dictionary has concept classes, such as interrogative expressions, negative expressions, and the kind of documents. The concept class corresponding to the kind of documents is introduced, because there are many e-mails, which request the sending of a catalog or a manual in the text class ''Request''. In the case of the address analysis task, the dictionary has concept classes such as tasks of departments, products which departments deal with, and words relating to the products. These dictionaries are used in the following experiments.

Table 2
Sizes of key concept dictionaries and their creation time

	Product	Contents	Address
Concept class	20	35	21
Key concept	235	96	226
Expression	427	259	475
Time (h)	17	24	37

4.3. Experimental methods

Keeping the frequency distribution of each class, we divide given e-mails into training ones and ones to be evaluated. The learning method generates training examples from the training e-mails and applies the examples to a fuzzy inductive learning algorithm. This method acquires a concept relation dictionary in the form of a fuzzy decision tree. On the other hand, the inference method generates examples to be evaluated from the e-mails to be evaluated and applies each example to the acquired concept relation dictionary. This method infers a text class for an e-mail corresponding to the example. We compared an inferred text class with a text class given by operators. If the inferred one was equal to the given one, the inference was judged to be correct. Lastly, we calculated the precision ratio p defined by Formula (2). In the field of information retrieval, recall ratio is also an important measure that indicates the effectiveness of retrieval systems. However, the inference method based on a fuzzy decision tree assigns each e-mail to only a text class. If we calculate the recall ratio based on the number of e-mails with the same text classes, the recall ratio is equal to the precision ratio. For example, each text class has 4 e-mails, 4 e-mails, and 2 e-mails in a three-classification task, respectively. Three e-mails, 1 e-mail, and 2 e-mails in the e-mails are correctly classified. Then, the precision ration is equal to 0.6 ($= \frac{3+1+2}{4+4+2}$) and the recall ratio is equal to 0.6 ($= \frac{4}{10} \times \frac{3}{4} + \frac{4}{10} \times \frac{1}{4} + \frac{2}{10} \times \frac{2}{2}$). We use only the precision ratio to evaluate the effectiveness of the analysis method.

$$p = \frac{\text{number of e-mails to be evaluated with right class}}{\text{number of e-mails to be evaluated}} \tag{2}$$

In the product analysis task and the contents analysis task, this experiment is performed 4 times by changing the number of training e-mails to 100, 200, 300, and 400. In the address analysis task, this experiment is performed 5 times by changing the number to 100, 200, 300, 400, and 500.

We also performed experiments based on SVM [2,16] to compare with the difference of classifier. In

the experiment, we used SVM software given by [6] and selected linear kernel and default parameters as its options. But, SVM regards a key concept as an attribute and regards whether a key concept appears or not as an attribute value, because SVM is unable to deal with a set value as an attribute value. We generate examples for SVM, acquire a concept relation dictionary from the examples in the form of hyperplanes, and evaluate the concept relation dictionary.

Moreover, we performed experiments in which a key concept dictionary was not used. That is, we broke down each e-mail into words by the lexical analysis and calculated the tf–idf value [14] for each word. We extracted words whose tf–idf value was bigger than or equal to a certain threshold. This method regards a word as an attribute, whether a word appears or not as an attribute value, and acquires concept relation dictionaries by using IDTF. The concept relation dictionary is evaluated. We performed preparatory experiments to decide the threshold before the experiments based on words. We generated concept relation dictionaries from small training e-mails by changing thresholds and selected a threshold corresponding to the concept relation dictionaries with the highest precision ratio. In the experiment, the threshold was 0.005. The number of extracted words is 1945 in the product analysis task and the contents analysis task, and the number is 2297 in the address analysis task.

4.4. Experimental results

Figs. 3–5 show trends in the precision ratios according to number of training examples. In these figures, the *x*-axis shows the number of training examples and the *y*-axis shows the precision ratio.

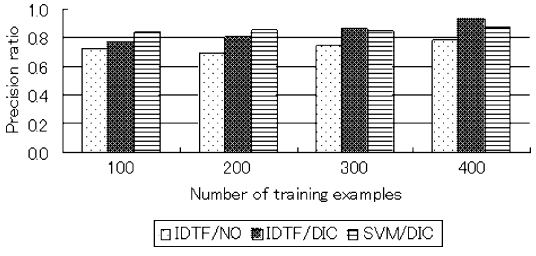

Fig. 3. Precision ratios in the product analysis task.

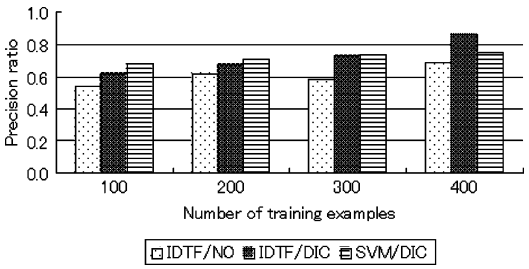

Fig. 4. Precision ratios in the contents analysis task.

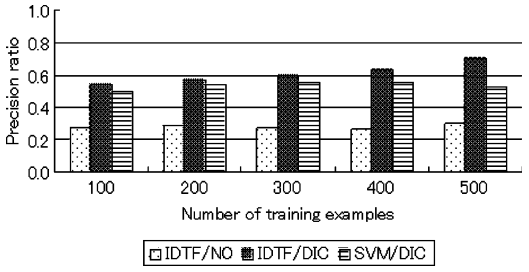

Fig. 5. Precision ratios in the address analysis task.

"IDTF/NO", "IDTF/DIC", and "SVM/DIC" show the results in each case. Here, "IDTF" shows the case where concept relation dictionaries are acquired by using IDTF and "SVM" shows the case where they are acquired by using SVM. Also, "DIC" shows the case where the key concept dictionaries are used and "NO" shows the case where they are not used.

4.5. Discussions

4.5.1. Acquired concept relation dictionaries

In the products analysis task, IDTF selects attributes corresponding to the model number of products and their functions as upper attributes. It acquires rules, such as "If the model number of products corresponding to a refrigerator is given, then a text class is Refrigerator" and "If the model number of products is not given and the function of products is sweeping, then a text class is Vacuum cleaner". In the contents analysis task, it selects attributes corresponding to documents as the top attribute. It acquires a rule, such as "If a kind of documents is a catalog and sending of the catalog is requested, then a text class is Request". In the address analysis task, it selects attributes corresponding to tasks of departments and the product, which departments deal with as upper

attributes. These selections and these rules correspond to the intuition of operators in our customer center. So, we think that the acquired concept relation dictionaries are valid.

4.5.2. Analysis tasks

The address analysis task is more difficult than the other two analysis tasks, because the address analysis task has 13 text classes and the number of e-mails included in each class is smaller than in the case of the other two tasks. Therefore, the precision ratios in the address analysis task are worse than the other ones. In addition, it takes a lot of time to create the key concept dictionary for the address analysis task.

The product analysis task and the contents analysis task have five text classes. However, the contents analysis task has a greater variant distribution of e-mails for the text classes. On the other hand, in the case of the product analysis task, extracted lexicons are apt to directly connect with a text class, because it is possible for the lexical analysis to extract the model number of products, the name of products, and so on. In the case of the contents analysis task, combined lexicons are apt to connect with a text class, because in most Japanese sentences the last lexicon determines the kind of sentence, i.e., an interrogative sentence or a negative sentence, and the combination of the front lexicons and the last lexicon is more important. The contents analysis task is more difficult than the product analysis task. Therefore, the contents analysis task has worse precision ratios and it takes more time to create a key concept dictionary for the task.

4.5.3. Number of training examples

Most of the precision ratios improve in each analysis task as the number of training examples increases. The experimental results show that the trends do not always converge. Precision ratios may further improve when we use more training examples. However, we could not use other e-mails, because we had to exclude personal information from them. In the future, we will use more e-mails and confirm whether the ratios are improved.

4.5.4. Key concept dictionaries

It is possible for a key concept dictionary to regard words and phrases, which have different expressions but the same meaning, as a key concept and to extract important key concepts for a task. Also, it is possible to sum up relevant key concepts by using a concept class. On the other hand, in the case of word selection based on the tf–idf value, there are about 2000 words and the number is much larger than the number of training examples. Inappropriate words are apt to be selected in the learning phase. Thus, the precision ratios in the case of using the key concept dictionaries are much higher than those in the word selection.

Regarding other aspects of the effectiveness of using the key concept dictionaries, we note the number of attributes. The number of attributes when using the key concept dictionaries is much smaller than that in the word selection. Therefore, fast learning is possible in the former case. On the other hand, the inference time when using the dictionary is almost equal to that in the case of the word selection, because the time for the lexical analysis occupies most of the inference time.

4.5.5. Classifier

We note the results in the case where number of training examples is the largest. IDTF using a key concept dictionary gives higher precision ratio than SVM using the dictionary does. SVM uses only the information of key concepts included in the dictionary, but IDTF uses the information of both concept classes and key concepts. It is possible for IDTF to get more information from e-mails. So, IDTF is able to acquire more accurate concept relation dictionaries. If SVM is able to use the information of concept classes, SVM may give better precision ratios. In the future, we will consider a method in which SVM deals with a set of key concepts as an attribute value.

IDTF acquires a concept relation dictionary in the form of a fuzzy decision tree. SVM acquires a concept relation dictionary in the form of hyper-planes. The dictionary by IDTF has high readability, but the dictionary by SVM does not have readability. A human expert is able to adjust the dictionary by IDTF. The adjustment will be able to lead to a higher precision ratio.

In the method based on IDTF, the effectiveness of fuzzy set is limited to the process of a key concept occurring only in the inference phase. However, textual data has much ambiguity. If we are able to process the ambiguity by fuzzy sets, the process will be able to lead to a higher precision ratio.

4.5.6. Evaluation by our customer center

In the case of the maximum training examples, the precision ratio of the product analysis task is 93.2%, the ratio of the contents analysis task is 86.3%, and the ratio of the address analysis task is 70.4%. According to the evaluation by Toshiba Corp.'s customer center, the level of the results for both the product analysis task and the contents analysis task is sufficient. The customer center considers the automatic classification for the contents analysis task to be particularly attractive, because the analysis task is apt to depend on the operator and the automatic classification is able to exclude the dependency.

However, according to the evaluation by the customer center, the level of the results for the address analysis task is insufficient. For automatic classification in the customer center, it will be necessary to improve the precision ratio. We intend to perform experiments using more training examples, because the number of training examples used was very small for 13 text classes. Also, we intend to refine the key concept dictionary by adding important words and phrases with human experts' help and to adjust the acquired concept relation dictionaries by the human experts. We think that these improvements will lead to a higher precision ratio. In the experiments, the inference method decides a text class with the maximum degree of certainty. However, it is possible for the inference method to automatically classify e-mails in cases where there is a high degree of certainty. In cases where there is a low degree of certainty, an operator can be used for classification. This semi-automatic method may be efficient for the customer center.

In view of the above discussion, we think that the proposed method is efficient for the analysis of e-mails at the customer center.

5. Conclusion

This paper proposes a method that automatically analyzes e-mails by using text mining techniques. Also, we applied the method to three kinds of analysis tasks: a product analysis task, a contents analysis task, and an address analysis task. We showed that the acquired concept relation dictionaries corresponded to the intuition of the operators in our customer center and gave high precision ratios in the classification.

In the future, we intend to improve performance through the addition of training examples, by revising the key concept dictionaries, by adjusting the acquired concept relation dictionaries, and by semi-automatic classification based on the degree of certainty, in order to enable use in real-world environments. On the other hand, we are developing tools supporting the creation of a key concept dictionary in order to reduce the creation time. However, since a human expert still has to create the dictionary through trial and error, the creation requires a lot of time. Therefore, we are planning to consider a method that automatically adjusts the key concept dictionary and analyzes e-mails without using a key concept dictionary

References

[1] C. Apte, Text Mining Applications for the Electronic Help Desk, in: Proceedings of 4th International Conference and Exhibition on the Practical Application of Knowledge Discovery and Data Mining, 200019–25.

[2] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machine and other Kernel-based Learning Methods, Cambridge, 2000.

[3] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407.

[4] R. Feldman, I. Dagan, H. Hirsh, Mining text using keyword distributions, J. Intell. Inf. Syst. 10 (1998) 281–300.

[5] T. Hofmann, Probabilistic Latent Semantic Indexing, in: Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval, 199950–57.

[6] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A Practical Guide to Support Vector Classification, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] Y. Ichimura, Y. Nakayama, M. Miyoshi, T. Akahane, T. Sekiguchi, Y. Fujiwara, Text mining system for analysis of a Salesper son's daily reports, in: Proceedings of Pacific Association for Computational Linguistics 2001, 2001127–135.

[8] J. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Technical Report LS-8 Report 23, Computer Science Department, University of Dortmund, 1997.

[9] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1992.

[10] J. Rennie, ifile: an application of machine learning to e-mail filtering, in: Proceedings of KDD 2000 Workshop on Text Mining, 2000.

[11] B. Raskutti, H. Ferra, A. Kowalczyk, Combining clustering and co-training to enhance text classification using unlabelled data, in: Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002620–625.
[12] S. Sakurai, Y. Ichimura, A. Suyama, R. Orihara, Acquisition of a knowledge dictionary for a text mining system using an inductive learning method, in: Proceedings of IJCAI 2001 Workshop on Text Learning: Beyond Supervision, 200145–52.
[13] S. Sakurai, Y. Ichimura, A. Suyama, Acquisition of a knowledge dictionary from training examples including multiple values, in: Proceedings of 13th International Symposium, ISMIS 2002, 2002103–113.
[14] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.
[15] P.-N. Tan, H. Blau, S. Harp, R. Goldman, Textual data mining of service center call records, in: Proceedings of 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000417–423.
[16] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
[17] O. De Vel, Mining e-mail authorship, in: Proceedings of KDD 2000 Workshop on Text Mining, 2000.