Acquisition of a knowledge dictionary for a text mining system using an inductive learning method

Shigeaki Sakurai and Yumi Ichimura and Akihiro Suyama and Ryohei Orihara
Corporate Research & Development Center, TOSHIBA Corporation
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, Kanagawa, 212-8582, Japan

Abstract

A text mining system using domain-dependent dictionaries efficiently analyzes text data. The dictionaries store not only important words for the domains, but also rules composed of some important words. This paper proposes a method that automatically acquires the rules from the text data and their classes by using an inductive learning method. The paper shows that a fuzzy inductive learning algorithm is more appropriate for the acquisition of the rules than a crisp one. Also, it shows that the method acquires rules with high accuracy through numerical experiments based on 10-fold cross validation and using daily business reports in retailing.

1 Introduction

Although large amounts of information are stored on computers and used, our ability to process the information is limited and it is not always possible to process large amounts of the information efficiently.

This is the context in which data mining techniques have been studied with a view to using the information efficiently [Agrawal and Strikant, 1994][Terano and Ishino, 1996]. Using the techniques, it is possible to classify the information and find rules hidden in it. These techniques employ inductive learning and statistical approaches. Also, these techniques are used in combination with OLAP tools; the tools edit the information and visualize it. The techniques are premised on processing the information presented in the form of a tabular structure.

However, the information is not restricted to the form, and the information presented in the form of a free text is also stored. Text mining techniques [Tan et al., 2000][Feldman et al., 1998][Ichimura et al., 2000] have recently been studied as techniques that process the information. The technique shown in [Tan et al., 2000] characterizes service call center records by clustering based on keywords and key phrases and extracts relations between characters and a class given by a human. In this technique, it is difficult to give a meaning to a feature generated by clusters. On the other hand, the second one in [Feldman et al., 1998] shows relations between topics and trends of a topic in news articles by using a keyword dictionary and co-occurrence frequencies of keywords, where the dictionary describes hierarchical relations between keywords. The third one in [Ichimura et al., 2000] classifies daily business reports by using two kinds of knowledge dictionaries and realizes to read only the reports included in a class of interest, not to read all the reports. In these techniques, the dictionaries are important and have a great influence on results. However, the goodness of the dictionaries usually depends on an effort of a human expert. Thus, it is necessary to establish such a method that creates them, in order to process many kinds of problems.

In this paper, an attempt is automatically made to acquire one of the dictionaries. The paper proposes a learning method for the dictionary based on an inductive learning algorithm and an inference method in conjunction with the learning method. Also, the paper shows that a fuzzy inductive learning algorithm IDF [Sakurai and Araki, 1996] is appropriate for the methods through numerical experiments based on 10-fold cross validation, and using daily business reports in retailing.

2 Text mining system

In this section, the text mining system proposed in [Ichimura et al., 2000], is briefly reviewed.

2.1 Process flow

The text mining method classifies daily business reports by using two kinds of knowledge dictionaries. One is called the key concept dictionary. The other is called the concept relation dictionary. In the next section, the dictionaries are explained in detail. The method decomposes a daily business report into words by lexical analysis. Each word is checked as to whether the word is registered in the key concept dictionary. If a word is registered in the dictionary, a key concept corresponding to the word is assigned to the report. The report is also checked as to whether there is a key concept corresponding to the word is assigned to the report. The report is also checked as to whether there is a key concept described in the concept relation dictionary. If the report has the set, a text class corresponding to the set is assigned to the report. The method puts reports that have the same text class into a group. That is, the method
classifies the reports according to the process flow shown in Figure 1.

Figure 1: Flow of the text mining method

Therefore, it is enough for each reader to read only reports classified in classes, which represent their interests. At the same time, the system allows them to follow the trend of the key concepts for all reports. The reader can easily and efficiently use the information.

2.2 Knowledge dictionary

The key concept dictionary is composed of three layers. The first layer is called the concept class and shows a set of concepts that have a common feature. The second layer is called the key concept and shows a set of expressions that have the same meaning. The third layer is called the expression and shows important phrases and words concerning a target problem. In this layer, the phrases and the words are described in consideration with the variety of words. For example, Figure 2 shows a key concept dictionary in retailing. In this figure, “Sales” is a concept class; “Good sales”, “Bad sales”, and “V.S. last year” are key concepts in the concept class “Sales”; “sell out”, “sell well”, and “sales improve” are expressions in the key concept “Good sales”; and “remain unsold”, “do not sell well”, and “sales fall off” are expressions in the key concept “Bad sales”.

On the other hand, the concept relation dictionary is composed of a relation that has a condition part and a result part. A conjunction of key concepts is described in the condition part and a text class is shown in the result part. The text class shows a viewpoint that classifies reports concerning a target problem. For example, Figure 3 shows a relation described in a concept relation dictionary in retailing. The relation has a meaning such that if there are two key concepts “Put a sample” and “Good sales”, a text class is “Best practice”.

The dictionaries make it possible to classify the reports based on a consideration of the complicated concept relations.

The dictionaries are created through trial and error, a time-consuming process. Furthermore, the dictionaries must be created for each target problem. Therefore, the creation is a bottleneck for applying a text mining system to a new target problem. In the following section, we propose a method that automatically acquires the concept relation dictionary as one solution that overcomes the bottleneck. Also, we propose an inference method using the acquired dictionary.

3 An inductive learning of a concept relation dictionary

3.1 Learning method

This section proposes such a method that acquires a concept relation dictionary according to the process flow shown in Figure 4.

A relation stored in a concept relation dictionary shows meanings that occur by combining key concepts. The relation is regarded as a kind of rule. The rule is acquired by using an inductive learning method [Quinlan, 1992][Sakurai and Araki, 1996], if training examples are
available. Thus, we examine a method that generates the training examples as follows.

It is necessary for a training example to be composed of key concepts and a text class, because each rule is composed of them. On the other hand, if a human reads through a daily report, they understand the meaning of it and are able to attach an appropriate text class to it. Also, words are extracted from the report by lexical analysis. Each word is checked as to whether it corresponds to an expression in a key concept dictionary. If a word corresponds to an expression, a key concept corresponding to the expression and a concept class corresponding to the concept are extracted. Thus, if we can regard a concept class as an attribute, a key concept as an attribute value, and a text class given by the reader as a class for a training example, it is possible to generate a training example from a report. However, a specific attribute value “nothing” should be assigned to an attribute that does not have a corresponding attribute value, because a report does not always have descriptions corresponding to all concept classes. Figure 5 shows that a training example is created from a report.

Reports written by humans usually include fuzzy descriptions. It is necessary for an inductive learning method, which acquires a key concept relation dictionary, to process training examples that consist of attribute values with fuzziness. The IDF [Sakurai and Araki, 1996] shown in Table 1 does processing based on the fuzzy set theory. Thus, an IDF is adapted as an inductive learning algorithm incorporated in the learning method.

The IDF is similar to the C4.5[Quinlan, 1992]. But, the IDF uses a degree of certainty of each training example to calculate a gain_ratio and to decompose the training example subset allocated to an interior node. In step 4b, the IDF 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. The IDF calculates degrees for all examples included in the subset and calculates a gain_ratio by using the total of them instead of the number of the examples. Also, in step 4d, the IDF gathers examples, of that updated degrees are more than 0, for items corresponding to the selected attribute and generates a new subset corresponding to each item.

The IDF acquires rules from generated training examples, where the rules are described by using a fuzzy decision tree shown in Figure 6. In the fuzzy decision tree, each interior node described by a highlighted circle stores an attribute, each terminal node described by a shaded circle stores classes with a degree of certainty, each branch described by a line between nodes stores a fuzzy class item, and a path from the root node to a terminal node shows a rule. That is, the tree has three rules such as:

If “Sales” is “Bad Sales”, then the class is “Missed opportunity” with 1.0.
If “Sales” is “Good Sales” and “Reputation” is “Bad reputation”, then the class is “Other” with 0.8 and the class is “Best practice” with 0.2.
If “Sales” is “Good Sales” and “Reputation” is “Good reputation”, then the class is “Best practice” with 1.0.

This section showed how a training example set is created. Also, it showed an IDF is an appropriate inductive learning algorithm that acquires a concept relation dictionary. Thus, the process flow shown in Figure 4 successfully creates the dictionary, described by a fuzzy decision tree, from the reports and their corresponding
Table 1: IDF 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 terminal 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 [Quinlan, 1992] 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.

3.2 Inference method

For example, we examine the case in which the fuzzy decision tree shown in Figure 6 is given, and a report to be evaluated in which the attribute “Sales” has the attribute value “Good sales”, and the attribute “Reputation” has the attribute value “Normal reputation”. The report has 1.0 as the initial degree of certainty. But, the fuzzy decision tree does not have a branch corresponding to the value “Normal reputation”. If an inference method does not take fuzziness into consideration, the method cannot evaluate the report in the interior node. In such cases, the tree gives only an evaluation with “impossibility” or the default class. On the other hand, if the inference method takes fuzziness into consideration, the report should be transferred to the nodes immediately below with the equal degree of certainty given to each of these nodes. In each terminal node, the degree of certainty for the report and the degree of certainty for the text classes are evaluated. That is, “Other” is 0.4 ($=0.8 \times 0.5$) and “Best practice” is 0.1 ($=0.2 \times 0.5$) in the terminal node $T_2$, and “Best practice” is 0.5 ($=1.0 \times 0.5$) in the terminal node $T_3$. Finally, a text class corresponding to the report is decided by the total of the evaluations in each terminal node. In this case, “Best practice” is selected as a text class of the report, because the class has the maximum degree of certainty. Figure 7 shows how the inference taking into consideration fuzziness is performed.

Even if attribute values allocated to branches of an interior node are not equal to an attribute value in a report to be evaluated, the fuzzy inference method transfers the report to lower nodes by using their membership functions and calculating grades between the value and them. The method evaluates the report using another attributes in the lower nodes. On the other hand, the crisp inference method does not transfer it to the nodes.
and does not evaluate it using another attributes in such a case. Therefore, the fuzzy inference method utilizes attribute values from the report more efficiently than the crisp one does, thus the fuzzy one is more appropriate as an inference method for the application. In the following, this method is formalized.

In an interior node, the method calculates a grade for each subnode according to Formula (1).

If \( e_k = f_{n_i,j}, \exists j \),
then \[
\begin{align*}
&\text{grade}_{n_i,l} = 1, l = j \\
&\text{grade}_{n_i,l} = 0, l \neq j.
\end{align*}
\]
Otherwise,
\[
\text{grade}_{n_i,l} = 1, \forall l.
\]

(1)

Here, \( f_{n_i,j} \) is the \( j \)-th fuzzy class item corresponding to an interior node \( n_i \) with the \( k \)-th attribute, and \( e_k \) is the \( k \)-th attribute value in a report to be evaluated.

The method also transfers the report to be evaluated, which has a degree of certainty calculated by Formula (2), to each subnode, \( n_p \).

\[
\text{prob}_{n_p} = \frac{\text{prob}_{n_i}}{\sum_j \text{grade}_{n_i,j}}
\]

(2)

Here, \( \text{prob}_{n_i} \) is the degree of certainty for the report assigned to an interior node, \( n_i \).

In a terminal node, the degree of certainty corresponding to the transferred report is multiplied by the one corresponding to each class of the node. The degree is also summed up for each class. Namely, the degree of certainty corresponding to a class is calculated according to Formula (3).

\[
\text{rprob}_q = \sum_{n_s \in N_t} \text{prob}_{n_s} \times \text{cprob}_{n_s,q}
\]

(3)

Here, \( N_t \) is a set of terminal nodes and \( \text{cprob}_{n_s,q} \) is the degree of certainty corresponding to the \( q \)-th class in a terminal node, \( n_s \).

Thus, the inference method infers classes with the degree of certainty for a report to be evaluated. If it is necessary to select a class, the method selects the class with the maximum degree of certainty.

4 Numerical experiments

In this section, we show that the proposed learning method and inference method are efficient through numerical experiments in retailing.

4.1 Experimentation

Training examples

The text mining system [Ichimura et al., 2000] classifies daily business reports concerning retailing into three text classes: “Best practice”, “Missed opportunity”, and “Other”. In this system, the key concept dictionary is composed of the 13 concept classes and each concept class has a respective subset of key concepts. Table 2 shows the name of the concept classes and the number of the subset. The subset is designed so that key concepts in a subset are mutually exclusive. In numerical experiments, a subset is regarded as a concept class. Each attribute has either a key concept included in a subset or “nothing” as an attribute value. Also, a text class given by a concept relation dictionary is regarded as a class of a training example. In the meantime, the concept relation dictionary, which is created by a human and used as a reference in this experiment, has 349 rules. Thus, a training example is a composed of 142 attributes and its class.

Table 2: Number of subsets of key concepts

<table>
<thead>
<tr>
<th>Concept class</th>
<th>Subset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circulation</td>
<td>4</td>
</tr>
<tr>
<td>Display</td>
<td>20</td>
</tr>
<tr>
<td>Society</td>
<td>5</td>
</tr>
<tr>
<td>Shop</td>
<td>4</td>
</tr>
<tr>
<td>Promotion</td>
<td>21</td>
</tr>
<tr>
<td>Event</td>
<td>21</td>
</tr>
<tr>
<td>Demand</td>
<td>4</td>
</tr>
<tr>
<td>Price</td>
<td>14</td>
</tr>
<tr>
<td>Stock</td>
<td>19</td>
</tr>
<tr>
<td>Reputation</td>
<td>8</td>
</tr>
<tr>
<td>Trade</td>
<td>12</td>
</tr>
<tr>
<td>Sales</td>
<td>8</td>
</tr>
<tr>
<td>Comments</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>142</td>
</tr>
</tbody>
</table>

On the other hand, descriptions in the reports sometimes contain contradictions, and key concepts that cannot occur simultaneously can be extracted from the reports. The training examples based on such reports make a learned concept relation dictionary inaccurate. Thus, the reports are regarded as unnecessary reports. Training examples corresponding to them are not generated.

In the experiments, we utilized 1,044 daily business reports, and generated 780 training examples by deleting unnecessary reports. Here, the degree of certainty for each training example was equal to 1, because each text had only one text class. In Table 3, the number of generated examples for each class is shown.

Table 3: Number of training examples

<table>
<thead>
<tr>
<th>Text class</th>
<th>Number of examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best practice</td>
<td>47</td>
</tr>
<tr>
<td>Missed opport</td>
<td>50</td>
</tr>
<tr>
<td>Other</td>
<td>683</td>
</tr>
<tr>
<td>Total</td>
<td>780</td>
</tr>
</tbody>
</table>
Evaluation

The proposed methods are evaluated using 10-fold cross validation in order to avoid the bias of training examples. That is, the training example set is decomposed into 10 training example subsets. The learning method acquires a concept relation dictionary described by a fuzzy decision tree from 9 subsets. The inference method infers a text class for each training example included in the remaining subset. The inferred text class is compared with the text class preassigned to the training example. The learning and inference are repeated 10 times by replacing the subset used in the inference with a different subset. Lastly, the error ratio defined by Formula (4) is calculated.

\[
\text{Error ratio} = \frac{\text{Number of misclassified examples}}{\text{Number of evaluated examples}} \times 100 \tag{4}
\]

Section 3.1 qualitatively showed that the IDF is an appropriate inductive learning algorithm. In this section, in order to quantitatively verify the efficiency of the IDF, the learning and inference methods based on a crisp inductive learning algorithm, C4.5, are also tested. That is, the learning method in which the C4.5 is incorporated also acquires a concept relation described by a decision tree and the inference method infers a text class based on the decision tree. It is possible for the C4.5 to probabilistically process a numerical value in an interior node. However, the C4.5 regards a key concept as a discrete value and cannot process it probabilistically. Thus, the inference method does not take fuzziness into consideration in the decision tree. Instead of the consideration, the C4.5 generates both a branch and a subnode corresponding to each attribute value in an interior node, and allocates an appropriate text class to the subnode that does not have a training example. Thus, the C4.5 infers a text class for a report.

4.2 Experimental results

Table 4 shows the average size of 10 generated decision trees for the IDF and the C4.5. Here, the decision tree size is equal to the total of the number of interior nodes and the number of terminal nodes. Also, the table shows the learning error ratios and the evaluation error ratios for each case. The learning error ratio results are for the examples used in the learning that are applied to the tree generated by them, and the evaluation error ratio results are for the remaining examples.

<table>
<thead>
<tr>
<th>Table 4: Size and error ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Size</strong></td>
</tr>
<tr>
<td>Learning error ratio</td>
</tr>
<tr>
<td>Evaluation error ratio</td>
</tr>
</tbody>
</table>

On the other hand, Figure 8 shows the trend of error ratios accumulated in 10 numerical experiments. Here, the x-axis gives the number of the experiment, the y-axis gives the accumulated error ratios. Bar graphs with a slant line pattern show the ratios of the IDF, and bar graphs with a tartan pattern show the ratios of the C4.5. Also, each figure shows the trend corresponding to each text class. That is, Figure 8(a) shows the whole trend regardless the classes; Figure 8 (b) shows the trend in a class, “Best practice”; Figure 8 (c) shows the class, “Missed opportunity”; and Figure 8 (d) shows the class, “Other”.

Lastly, Figure 9 shows parts of a learned fuzzy decision tree using the IDF. In this figure, each interior node stores such a label that consists of a concept class and its subset, where a label is described as “word_a/word_b”, word_a shows a concept class, and word_b shows its subset. Also, each label on a line shows a key concept corresponding to the concept class and the subset described at its left side.

4.3 Discussion

This section shows that the learning method based on the IDF is more appropriate than the one based on the C4.5. Also, it shows that the proposed methods are efficient in terms of size of decision trees, error ratios, and selected attributes.

IDF vs. C4.5

Each fuzzy decision tree generated by the IDF has lower evaluation error ratios than each decision tree generated by the C4.5, as shown in Figure 8. The reason is that the IDF processes the fuzziness included in the reports in each interior node and terminal node. The result qualitatively verifies the efficiency of the IDF described in section 3.1.

Also, the size of the trees generated by the IDF is smaller than the one generated by the C4.5. In decision tree generation, the C4.5 generates branches and subnodes corresponding to all attribute values. On the other hand, the IDF generates only branches and subnodes that have reports corresponding to fuzzy class items. Therefore, the IDF generates more compact trees.

The result based on 10-fold cross validation avoids the bias included in the given examples. From the result, we conclude that the IDF is a more appropriate inductive learning algorithm for creating a concept relation dictionary than the C4.5.

Error ratio

In case of the IDF, the learning error ratio was 0.0% and the evaluation error ratio was 2.82%, as shown in Table 4. On the other hand, it is possible for a concept relation dictionary given by a human expert to completely infer right classes for given all report, because each report has such a class that is decided using the dictionary in the experiments. Thus, the generated fuzzy decision trees infer a text class for a report with mostly high accuracy. For respective classes, the classes “Best practice” and “Missed opportunity” are not as good as the class “Other”. The reason is the number of reports with the class “Other” is much greater than other classes, and
rules with the class “Other” tend to be preferred. However, it is possible for the IDF to give a weight for each example. This function of the IDF may give priority to the rules of the two classes. Thus, the difference in the error ratios between classes is not so important.

Size
Table 4 shows the average decision tree size is about 85 for the IDF. The number of rules included in the generated trees is smaller than 85, because the number of rules is equal to the number of terminal nodes. On the other hand, the number of rules is 349 in the text mining system. The generated rules infer a text class for a report with mostly high accuracy. Thus, we conclude that the learning method acquires a more compact concept relation dictionary.

Selected attributes
The learning method selects “Sales/profit” as the top attribute in all cases. This result verifies that the classes have a strong relation with “Sales/profit”. Thus, we can say that the method accurately acquires a concept relation dictionary in a representative way.

In summary, the learning method acquires a compact and accurate concept relation dictionary and the inference method infers an appropriate class for each evaluation example.

5 Summary and future works
The paper proposed a learning method that acquires a concept relation dictionary for a text mining system and an inference method using the dictionary. The paper also showed that a fuzzy inductive learning algorithm, IDF, is an appropriate algorithm for acquiring the dictionary through numerical experiments based on 10-fold cross validation, and using daily business reports in retailing.

As future works, we intend to acquire a key concept dictionary, because it is also time-consuming for a human to create the dictionary. We consider such a method that acquires the dictionary using a clustering method (e.g.
Figure 9: Parts of a learned fuzzy decision tree

[Agrawal et al., 1998], and such a method that does it using pre-defined relations among noun, adjective, adverb, and verb. Also, we will attempt to analyze a free text answer to a questionnaire. The analysis would be more difficult than the one concerning the business reports, because an answer to a questionnaire contains a large amount of data and is written in various ways. Moreover, we will attempt to process reports that have multiple attribute values. This paper used only reports that do not have them, but important information may be included in remaining reports. Fortunately, it is possible for the IDF to process the multiple values, if membership functions that express them are well defined. Thus, we intend to define appropriate membership functions in order to process them.

References


