An Empirical Approach to Text Categorization based on Term Weight Learning

Fumiyo Fukumoto and Yoshimi Suzuki†
Department of Computer Science and Media Engineering,
Yamanashi University
4-3-11 Takeda, Kofu 400-8511 Japan
{fukumoto@skye,ysuzuki@suwa†}.esi.yamanashi.ac.jp

Abstract

In this paper, we propose a method for text categorization task using term weight learning. In our approach, learning is to learn true keywords from the error of clustering results. Parameters of term weighting are then estimated so as to maximize the true keywords and minimize the other words in the text. The characteristic of our approach is that the degree of context dependency is used in order to judge whether a word in a text is a true keyword or not. The experiments using *Wall Street Journal* corpus demonstrate the effectiveness of the method.

Introduction

With increasing numbers of machine readable documents becoming available, an automatic text categorization which is the classification of text with respect to a set of pre-categorized texts, has become a trend in IR and NLP studies.

One of the important issues in text categorization task is how to characterize texts which are pre-categorized. There are at least two statistical approaches to cope with the issue, i.e. statistical approach that relies mainly on (1) *surface information* of words in texts, and (2) *semantic information* of words in texts.

Statistical approach based on surface information of words has been widely studied in IR. One representative is a vector model. In this model, each text is represented by a *vector*, i.e. every text which should be classified and texts which are pre-categorized in a training phase are characterized by a vector, each dimension of which is associated with a specific word in texts, and every coordinate of the text is represented by term weighting. Then, some similarity measure is used and the text is assigned to the most semantically similar set of texts which are pre-categorized. Term weighting method is widely studied [Luhn1958], [Salton and Yang1973], [Salton1988], [Jones1973]. Guthrie and Yuasa used word frequencies for weighting [Guthrie and Walker1994], [Yuasa et al.1995], and Tokunaga used weighted inverse document frequency (WIDF) which is a word frequency within the document divided by its frequency throughout the entire document collection [Tokunaga and Iwayama1994].

The other approach is based on a probabilistic model. This approach is widely used, since it has solid formal grounding in probability theory. Iwayama et. al. proposed a probabilistic model called *Single random Variable with Multiple Values* (*SVMV*) [Iwayama and Tokunaga1994]. They reported that the result of their experiment using *SVMV* was better than other probabilistic models; *Component Theory*(CT) [Kwok1989], *Probabilistic Relevance Weighting*(PRW) [Robertson and Jones1976] and *Retrieval with Probabilistic Indexing*(RPI) [Fuhr1989] in the task of categorizing news articles from the *Wall Street Journal*(*WSJ*). Most previous approaches seem to show the effect in entirely different texts, such as 'weather forecasts', 'medical reports' and 'computer manuals'. Because each different text is characterized by a large number of words which appear frequently in one text, but appear seldom in other texts. However, in some texts from the same domain such as 'weather forecasts', one encounters quite a large number of words which appear frequently over texts. Therefore, how to characterize every text is a serious problem in such the restricted subject domain.

The other statistical approach is based on semantic information of words. The technique developed by Walker copes with the discrimination of polysemy [Walker and Amsler1986]. The basic idea of his approach is that to disambiguate word-senses in articles might affect the accuracy of context dependent classification, since the meaning of a word characterizes the domain in which it is used. He used the semantic codes of the *Longman Dictionary of Contemporary English* to determine the subject domain for a set of texts. For a given text, each word is checked against the dictionary to determine the semantic codes associated with it. By accumulating the frequencies for these senses and then ordering the list of categories in terms of frequency, the subject matter of the text can be identified. However, Fukumoto reported that when using disambiguated word-senses within texts (49 different texts, each of which consists of 3,500 sentences) were up to only 7.5% as those when using word frequencies for

weighting, since in a restricted subject domain such as *Wall Street Journal*, lots of nouns in articles were used with the same sense. As a result, the results of word-sense disambiguation did not strongly contribute to an accurate classification [Fukumoto and Suzuki1996].

Blosseville et. al. proposed an automated method of classifying research project descriptions using textual and non-textual information associated with the projects. Textual information is processed by two methods of analysis: a NL analysis followed by a statistical analysis. Non-textual information is processed by a symbolic learning technique. The results using two classification sets showed that 90.6% for 7 classes and 79.9% for 28 classes could be classified correctly. Their method, however, requires a great effort, since the input data are not raw textual data, but rather the result of deep syntactic and semantic analysis of textual data.

In this paper, we propose an alternative method for an automatic classification, i.e. a method for term weight learning which is used to characterize texts. In our approach, learning is to learn true keywords from the error of clustering results. Parameters of term weighting are then estimated so as to maximize the true keywords and minimize the other words in the text. The characteristic of our approach is that the degree of context dependency is used in order to judge whether a word in a text is a true keyword or not. We applied our technique to the task of categorizing news articles from 1989 *WSJ* in order to see how our method can be used effectively to classify each text into a suitable category.

In the following sections, we first present a basic idea of context dependency, and describe how to recognize keywords. Next, we describe methods for term weight learning and for classifying texts using term weight learning. Then, we present a method for categorization task. Finally, we report on some experiments in order to show the effect of the method.

Training the Data

Recognition of Keywords

In our approach, learning is to learn true keywords from the error of clustering results. The basic idea of our term weight learning is to use the fact that whether a word is a key in a text or not depends on the domain to which the text belongs.

We will focus on the *WSJ* corpus. Let 'stake' be a keyword and 'today' not be a keyword in the text (article). If the text belongs to a restricted subject domain, such as 'Economic news', there are other texts which are related to the text. Therefore, the frequency of 'stake' and 'today' in other texts are similar to each other. Let us further consider a broad coverage domain such as all texts of the *WSJ*; i.e. the text containing the words 'stake' and 'today' belongs to the *WSJ* which consists of different subject domains such as 'Economic news' or 'International news'. 'Today' should appear frequently with every text even in such a domain, while 'stake' should not. Our technique for recognition of true keywords explicitly exploits this feature of context dependency of word: how strongly a word is related to a given context?

Like Luhn's assumption of keywords, our method is based on the fact that a writer normally repeats certain words (keywords) as he advances or varies his arguments and as he elaborates on an aspect of a subject [Luhn1958]. Figure 1 shows the structure of the *WSJ* corpus.

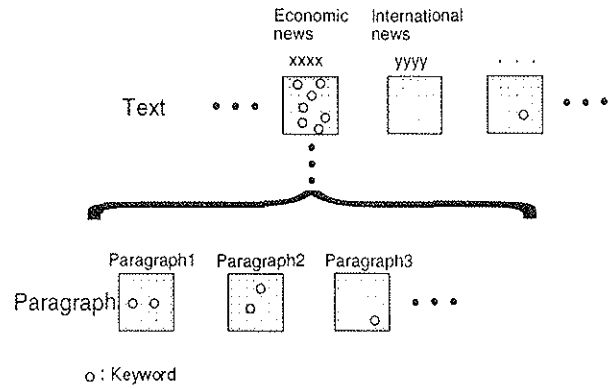

Figure 1: The structure of the *WSJ* corpus

In Figure 1, 'xxxx' and 'yyyy' shows a title name of a text which belongs to the category, 'Economic news' and 'International news', respectively.

We introduce a degree of context dependency into the structure of the *WSJ* corpus shown in Figure 1 in order to recognize keywords. A degree of context dependency is a measure showing how strongly each word is related to a particular paragraph or text. In Figure 1, let '◯' be a keyword in the text 'xxxx'. According to Luhn's assumption, '◯' frequently appears throughout paragraphs. Therefore, the deviation value of '◯' in the paragraph is small. On the other hand, the deviation value of '◯' in the text is larger than that of the paragraph, since in texts, '◯' appears in the particular text, 'xxxx'. We extracted keywords using this feature of the degree of context dependency. In Figure 1, if a word is a keyword in a given text, it satisfies that the deviation value of a word in the paragraph is smaller than that of the text, and is shown in formula (1) [Fukumoto et al.1997].

$$\frac{\chi P_w^2}{\chi T_w^2} < 1 \tag{1}$$

where,

$$\chi P_w^2 = \sqrt{\frac{\sum_{j=1}^{n}(\chi P_{wj}^2 - \bar{v}_w)}{w}} \tag{2}$$

$$\chi P_{wj}^2 = \frac{(x_{wj} - \bar{v}_{wj})^2}{\bar{v}_{wj}} \tag{3}$$

$$\bar{v}_{wj} = \frac{\sum_{j=1}^{n} x_{wj}}{\sum_{w=1}^{m}\sum_{j=1}^{n} x_{wj}} \times \sum_{w=1}^{m} x_{wj} \qquad (4)$$

In formula (1), w of χP_w^2 and χT_w^2 is a word in paragraph and text, respectively. χP_w^2 and χT_w^2 is the deviation value of a set of paragraph and text, respectively. In formula (2), n is the number of paragraphs, and $\bar{v}_w$ is the mean value of the total frequency of word w in paragraphs which consist of n. In formula (3), x_{wj} is the frequency of word w in the j-th paragraph. $\bar{v}_{wj}$ in formula (3) is shown in (4) where m is the number of different words and n is the number of paragraphs [1].

Term Weight Learning

In our method, non-overlapping group average clustering algorithm based on frequency-based term weighting is applied to every text which is pre-categorized. If a text which could not be clustered correctly in the process of clustering, then, recognition of keywords is performed.

Let T_x and $T_{x'}$ be the same category and T_y not be the same one with T_x. Let also T_x and T_y be judged to be the same category incorrectly. Recognition of keywords is shown in Figure 2.

In Figure 2, (a-1) and (b-1) are the procedures to extract keywords, and (a-2) and (b-2) are the procedures to extract other words. In (a), for example, when w is judged to be a keyword, term weighting of w is $\alpha \times f(w)$, where $f(w)$ is a frequency of w. On the other hand, when w is judged not to be a keyword, term weighting of w is $\beta \times f(w)$. Here, α and β is a variable which is concerned with a true keyword and the other words, respectively[2]. In $\frac{\chi P_w^2}{\chi T_w^2} < 1$ shown in Figure 2, the texts are T_x and T_y.

Clustering Texts based on Term Weight Learning

The clustering algorithm for pre-categorization of texts is shown in Figure 3.

As shown in Figure 3, the algorithm is composed of three procedures: **Make-Initial-Cluster-Set**, **Apply-Clustering** and **Term-Weight-Learning**[3].

1. **Make-Initial-Cluster-Set**

   The procedure **Make-Initial-Cluster-Set** produces all possible pairs of texts in the input with their similarity values. Firstly, every text which is the pre-categorization of texts is represented by a vector. Using a term weighting method, every text would be represented by a vector of the form

$$T_i = (X_{i1}, X_{i2}, \cdots, X_{ix}) \qquad (5)$$

   where x is the number of nouns in a text and X_{ij} is a frequency with which the noun X_j appears in text T_i.

   Given a vector representation of texts T_1, $\cdots$, T_m (where m is the number of texts) as in formula (5), a similarity between two texts T_i and T_j would be obtained by using formula (6). The similarity between T_i and T_j is measured by the inner product of their normalized vectors and is defined as follows:

$$Sim(T_i, T_j) = \frac{T_i \cdot T_j}{|\,T_i\,||\,T_j\,|} \qquad (6)$$

   The greater the value of $Sim(T_i, T_j)$ is, the more similar T_i and T_j. For texts T_1, $\cdots$, T_{m-1} and T_m, we calculate the similarity value of all possible pairs of texts. The result is a list of pairs which are sorted in the descending order of their similarity values. The list is called ICS (Initial Cluster Set). In the FOR-loop in the algorithm, a pair of texts is retrieved from ICS, one at each iteration, and passed to the next two procedures.

2. **Apply-Clustering**

   In this procedure, the clustering algorithm is applied to the sets and produces a set of clusters, which are ordered in the descending order of their semantic similarity values. We adopted non-overlapping group average method in our clustering technique [Jardine and Sibson1968]. Let T_x and $T_{x'}$ be the same category and T_y not be the same one with T_x. Let also T_x and T_y be judged to be the same category incorrectly. The next procedure, **Term-Weight-Learning** is applied to T_x, $T_{x'}$ and T_y.

3. **Term-Weight-Learning**

   For T_x, $T_{x'}$ and T_y ($T_{y'}$), recognition of keywords shown in Figure 2 is applied, and every text would be represented by a vector of the form

$$T_i = (X'_{i1}, X'_{i2}, \cdots, X'_{ix}) \qquad (7)$$

```
begin
   do Make-Initial-Cluster-Set
   for i := 1 to m(m-1)/2 do
       do Apply-Clustering
       if T_x such that T_x does not belong to
          the correct cluster
          then do Term-Weight-Learning
                  do Make-Initial-Cluster-Set
                  i := 1
       end_if
   end_for
end
```

Figure 3: Flow of the algorithm

[1] In formulae (2), (3) and (4), we can replace χP_w^2 with χT_w^2.

[2] In the experiment, two procedures are performed alternatively; (1) increment value of α is set to 0.001 and β is a constant value, (2) decrease value of β is set to 0.001 and α is a constant value.

[3] The largest value of α is empirically determined.

```
begin
(a) if T_y' such that T_y' and T_y be the same category exists
      for all w such that T_x ∩ T_y
        if w satisfies χP_w^2 / χT_w^2 < 1 and w is the element of T_x ∩ T_x' or T_y ∩ T_y'
(a-1)     then w is judged to be a keyword and parameter of term weighting of w is set to α (1 < α < 10)
        else if w does not satisfy χP_w^2 / χT_w^2 < 1 and w is the element of T_x ∩ T_x' or T_y ∩ T_y'
(a-2)     then w is judged not to be a keyword and parameter of term weighting of w is set to β (0 < β < 1)
        end_if
      end_for
(b) else
      for all w such that T_x ∩ T_y
        if w satisfies χP_w^2 / χT_w^2 < 1 and w is the element of T_x ∩ T_x'
(b-1)     then w is judged to be a keyword and parameter of term weighting of w is set to α (1 < α < 10)
        else if w does not satisfy χP_w^2 / χT_w^2 < 1 and w is the element of T_x ∩ T_x'
(b-2)     then w is judged not to be a keyword and parameter of term weighting of w is set to β (0 < β < 1)
        end_if
      end_for
    end_if
end
```

Figure 2: Recognition of keywords

where x is the number of nouns in a text and X'_{ij} is as follows;

$$X'_{ij} = \begin{cases} 0 & X'_j \text{ does not appear in } T_i \\ \alpha \times f(X'_j) & X'_j \text{ is a keyword and appears in } T_i \\ \beta \times f(X'_j) & X'_j \text{ is not a keyword and appears in } T_i \end{cases}$$

where $f(X'_j)$ is a frequency with which the noun X'_j appears in text T_i.

α and β are estimated so as to maximize $Sim(T_x, T_{x'})$ and $Sim(T_y, T_{y'})$ among all possible pairs of texts, T_x, $T_{x'}$, T_y and $T_{y'}$.

Make-Initial-Cluster-Set where every text except T_x, $T_{x'}$, T_y and $T_{y'}$ would be represented by a vector of the form shown in formula (5) and T_x, $T_{x'}$, T_y and $T_{y'}$ would be represented by a vector shown in formula (7), is applied to an arbitrary pair in texts, and the procedures are repeated.

If the newly obtained cluster contains all the texts in input, the whole process terminates.

Category Assignment

For the training data, $T_1, \cdots, T_m$ (where m is the number of texts), clustering algorithm which is shown in Figure 3 is applied, and all texts are classified into a suitable category. Given a new text T which should be classified, T would be represented by a term vector of the form shown in formula (5). The similarities between T and each text of the training data are calculated by using formula (6). Then, $T_1, \cdots, T_m$ are sorted in the descending order of their similarity values. T is assigned to the categories which are assigned to $T_1, \cdots, T_m$ with the descending order of their similarity values.

Lewis proposed the *proportional assignment strategy* based on the probabilistic ranking principle [Lewis1992]. Each category is assigned to its top scoring texts in proportion to the number of times the category was assigned in the training data. For example, a category assigned to 2% of the training texts would be assigned to the top scoring 0.2% of the test texts if the proportionality constant was 0.1, or to 10% of the test texts if the proportionality constant was 5.0. We used this strategy for evaluation.

Experiments

We have conducted two experiments to examine the effect of our method. The first experiment, **Text Categorization Experiment** shows how the results of term weight learning can be used effectively to categorize new texts. The second experiment, **Comparison to Other Methods**, we applied *chi-square* method as a *vector model* and Iwayama's *SVMV* as a *probabilistic model* to classify texts [Iwayama and Tokunaga1994], and compared them with our method.

Data

The training data we have used is 1989 *Wall Street Journal* (*WSJ*) in ACL/DCI CD-ROM which consists of 12,380 texts [Liberman1991]. The *WSJ* are indexed with 78 categories. Texts having no category were excluded. 8,907 texts remained. Each having 1.94 categories on the average. The largest category is "Tender Offers, Mergers, Acquisitions (TNM)" which encompassed 2,475 texts; the smallest one is "Rubber (RUB)",

assigned to only 2 texts. On the average, one category is assigned to 443 texts. All 8,907 texts were tagged by the tagger [Brill1992]. We used nouns in the texts. Inflected forms of the same words are treated as single units. For example, 'share' and 'shares' are treated as the same unit. We divided 8,907 texts into two sets; one for training(4,454 texts), and the other for testing(4,453 texts).

Text Categorization Experiment

Term weight learning is applied to 4,454 texts, and each word in the texts was weighted. For the result, we applied category assignment to the 4,453 test data. The best known measures for evaluating text categorization models are *recall* and *precision*, calculated by the following equations [Lewis1992].

$$Recall = \frac{\text{the number of categories that are correctly assigned to texts}}{\text{the number of categories that should be assigned to texts}}$$

$$Precision = \frac{\text{the number of categories that are correctly assigned to texts}}{\text{the number of categories that are assigned to texts}}$$

Note that recall and precision have somewhat mutually exclusive characteristics. To raise the recall value, one can simply assign many categories to each text. However, this leads to a degradation in precision, i.e. almost all the assigned categories are false. A *breakeven* point might be used to summarize the balance between recall and precision, the point at which they are equal. We calculated breakeven points in the experiment. The result of **Text Categorization Experiment** is shown in Table 1.

Table 1: The result of the experiment

Category	Training data	Test data	Breakeven
10	2,399	1,457	0.80
20	3,893	2,452	0.77
30	5,178	3,508	0.77
40	5,828	3,994	0.76
50	7,344	4,998	0.77
60	8,475	5,976	0.76
70	11,489	6,148	0.75
78	11,649	7,305	0.75

In Table 1, 'Category' shows the number of categories which are extracted at random. 'Training data' shows the number of training texts which are included in each category shown in the 'Category'. Most of the texts in *WSJ* are classified into more than one category. Each having 1.94 categories on the average. 'Test data' in Table 1 shows the total number of the texts which is classified into 'Category'.

Comparison to Other Methods

We reported on the results of our method comparing with other two methods, i.e. chi-square value for term weighting and *Single random Variable with Multiple Values(SVMV)* which is proposed by Iwayama et.al. [Iwayama and Tokunaga1994].

The reason why we compared our method with chi-square method is the following two points:

- Chi-square value is one of the conventional text classification [Iwadera and Kikui1997].
- In our method, chi-square value is used in order to introduce a degree of context dependency.

Iwayama et. al. proposed a new probabilistic model for text categorization called *SVMV*. The probability that the document d is classified into the category c is shown in formula (8).

$$P(c \mid d) = P(c) \sum_{t_i} \frac{P(T = t_i \mid c) P(T = t_i \mid d)}{P(T = t_i)} \quad (8)$$

where,

- $P(T = t_i \mid c) = \frac{NC_i}{NC}$: NC_i is the frequency of the term t_i in the category c, and NC is the total frequency of terms in c.
- $P(T = t_i \mid d) = \frac{ND_i}{ND}$: ND_i is the frequency of the term t_i in the document d, and ND is the total frequency of terms in d.
- $P(T = t_i) = \frac{N_i}{N}$: N_i is the frequency of the term t_i in the given training documents, and N is the total frequency of terms in the training documents.
- $P(c) = \frac{D_c}{D}$: D_c is the frequency of documents that is categorized to c in the given training documents, and D is the frequency of documents in the training documents.

They reported that in their experiment using *WSJ*, the result of the breakeven points of TF*IDF which was proposed by Salton et. al. was 0.48, while the result of *SVMV* was 0.63. Furthermore, their method is similar to our technique when the following two points are considered:

- Text categorization is defined as the classification of texts with respect to a set of pre-categorized texts.
- Category assignment is based on surface information of words in texts.

Therefore, we implemented Iwayama et. al.'s method and compared it with our method. The results are shown in Figure 4.

Figure 4 shows the recall/precision trade off for each method with proportional assignment strategy. 'learning', '*SVMV*' and 'χ^2' shows the result of our method, Iwayama's method and χ^2 value, respectively. Table 2 lists the breakeven points for each method. All the breakeven points were obtained when proportionality constant was about 1.0.

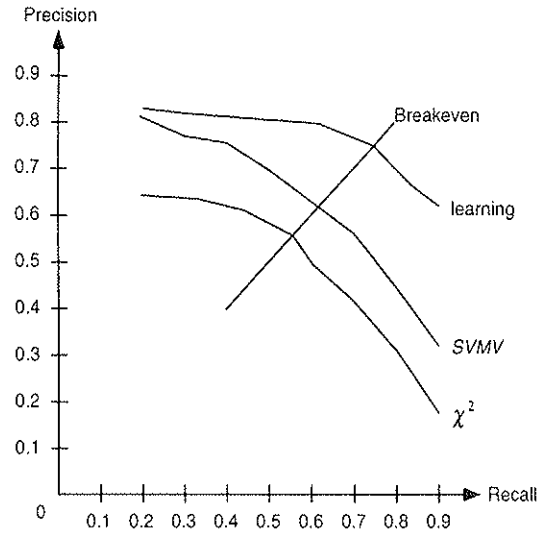

Figure 4: The result of comparative experiment

Table 2: Breakeven Points

Method	Breakeven Points
learning	0.75
SVMV	0.64
χ^2	0.56

Discussion

Text Categorization Experiment

Effectiveness of the Method According to Table 1, there are 7,305 test data in all which are classified into 78 categories, and the value of the breakeven points was 0.75. Comparing the ratios of correct judgments when the number of categories is large with when the number of it is small, the correctness of the former was higher in some cases. For example, when the number of categories was 40, the correct ratio was 0.76, while the number of categories was 50, the correct ratio was 0.77. This shows that our method can be used effectively to characterize each text without depending on the number of categories.

Table 3 shows the first top five of the highest weighted value of 12 categories which were selected from 78 categories at random.

In Table 3, 'Word' shows the extracted words, and 'Wt' shows its weighted value. 12 categories which are used in Table 3 are shown in Table 4.

According to Table 3, our technique for term weight learning is effective, though there are some nouns judged highly weighted but our intuition cannot explain why. For example, 'general' in 'FOD' is not a true keyword in our intuition.

Table 4: The category name

AIR:	Airlines	ARO:	Aerospace
BBK:	Buybacks	BNK:	Banks
FOD:	Food products	STK:	Stock market
ENV:	Environment	MED:	Media
ECO:	Economic news	PIP:	Pipeline
DIV:	Dividends	CPR:	Computers

Problem of the Method The test data which was the worst result, was the data which should be classified into 'STK'. There were 499 test data which should be classified into 'STK'. Of these, 159 data (32% in all) be judged to classify into 'BBK', incorrectly. According to Table 3, the first top three words in 'BBK' and those of 'STK' are the same, and the weighted values of these words of 'BBK' are higher than those of 'STK'. 'BBK' and 'STK are semantically similar with each other and it is difficult to distinct even for a human. Therefore, in this case, there are limitations to our method using term weight learning.

Comparison to Other Methods

(1) χ^2 method and our method Table 2 shows that the breakeven points using our method was 0.75, while χ^2 was 0.56. Table 5 shows the first top five of the highest weighted value of 12 categories using χ^2 method.

According to Table 5, every noun except 'devon' and 'hadson' in 'BBK' and 'transcanada' and 'westcoast' in 'PIP' are correctly weighted as keywords in every categories. On the other hand, the test data which was the worst result, was the same data as the result using our method, i.e. the data which should be classified into 'STK'. According to Table 5, three words in 'BBK' and those of 'STK' are the same, and the weighted values of these words of 'STK' are higher than those of 'STK'. As a result, it is difficult to distinct these two categories in χ^2 method.

One possible reason why the result of our method was better than χ^2 method is that the difference between weighting values of two words in χ^2 was smaller than those of our method. The deviation value between an arbitrary two keywords in both methods is shown in Table 6.

Table 6: Deviation value of χ^2 and our methods

Cat.	learning	χ^2	Cat.	learning	χ^2
AIR	4.63	3.64	ARO	4.20	4.12
BBK	3.80	2.57	BNK	2.23	2.25
FOD	2.25	2.72	STK	4.45	2.57
ENV	2.99	2.30	MED	3.89	6.10
ECO	4.44	2.55	PIP	3.94	3.11
DIV	4.93	3.41	CPR	4.50	3.86

Table 3: The first top 5 of the highest weighted words in our method

	AIR		ARO		BBK		PIP	
No	Word	Wt	Word	Wt	Word	Wt	Word	Wt
1	airline	522.1	aerospace	148.2	share	149.0	gas	58.0
2	mile	136.5	aircraft	143.0	stock	71.9	pipeline	37.0
3	passenger	120.5	air	730.	company	57.2	industry	29.0
4	revenue	85.0	army	51.0	bank	51.0	foothill	24.0
5	air	67.2	jetliner	43.3	security	43.5	oil	7.0
	BNK		FOD		STK		DIV	
No	Word	Wt	Word	Wt	Word	Wt	Word	Wt
1	bank	84.0	food	140.0	company	50.0	cent	85.0
2	branch	32.0	fda	27.0	share	37.7	share	70.0
3	credit	30.0	general	24.0	stock	31.7	company	60.9
4	tax	24.0	cereal	19.0	trade	10.1	dividend	54.6
5	letter	16.0	health	16.0	investment	9.4	split	46.7
	ENV		MED		ECO		CPR	
No	Word	Wt	Word	Wt	Word	Wt	Word	Wt
1	environment	78.0	news	281.0	gain	120.5	analytics	106.5
2	maquilas	19.0	d&b	108.0	tax	111.0	IBM	89.8
3	water	12.0	network	69.1	capital	83.4	machine	69.0
4	plant	10.1	report	69.0	rate	79.5	computer	62.0
5	health	9.4	broadcaster	44.8	economy	30.5	system	48.6

Table 5: The first top 5 of the highest weighted words in χ^2 method

	AIR		ARO		BBK		PIP	
No	Word	Wt	Word	Wt	Word	Wt	Word	Wt
1	airline	12109.1	boeing	4880.0	share	2348.7	pipeline	8521.7
2	ual	5268.5	force	4022.3	redemption	1902.4	foothill	5933.7
3	passenger	5142.3	aircraft	3886.7	devon	1779.4	gas	5744.4
4	pilot	4672.1	defense	2328.6	hadson	1641.1	transcanada	4948.0
5	flight	4050.8	missile	2060.7	buy-back	1616.4	weastcoast	4494.9
	BNK		FOD		STK		DIV	
No	Word	Wt	Word	Wt	Word	Wt	Word	Wt
1	bank	6196.4	spam	3148.4	stock	7265.4	dividend	10067.7
2	bnl	1517.3	food	2848.5	share	3563.2	share	4999.4
3	bond	1211.4	cereal	2627.7	buy-back	2302.0	company	3666.8
4	loan	1023.3	cholesterol	2518.2	redemption	1448.5	buy-back	2499.4
5	rate	890.1	cooke	2355.1	big	1018.6	henley	2166.6
	ENV		MED		ECO		CPR	
No	Word	Wt	Word	Wt	Word	Wt	Word	Wt
1	ozone	2650.7	magazine	4222.3	gain	2160.5	computer	13948.8
2	epa	2414.0	d&b	3313.7	democrat	1492.0	IBM	8470.1
3	asbestosis	2259.0	cable	2890.1	tax	1410.6	software	4709.2
4	anthrax	1483.5	network	2496.9	budget	1294.5	cray	3538.7
5	pollution	1165.3	broadcaster	1999.9	spending	1157.3	digital	3291.7

In Table 6, the deviation value using χ^2 method was smaller than our method except 'BNK', 'FOD' and 'MED'. This shows that χ^2 method can not represent the characteristic of the text more precisely than our method.

(2) *SVMV* method and our method According to Table 4, the breakeven points using our method was 0.75, while *SVMV* was 0.64, respectively.

A possible reason why the result of our method was better than *SVMV* is that term weight learning is effective to classify texts. Let A and B be a category name and the total number of words which were included in each category be the same. Let also w_1 is included in A, B and the test data with the same frequency, and the test data consists of only w_1. In *SVMV*, the probabilities of the test data which is classified into A and B are the same. Therefore, it could not be judged whether the test data is classified into A or B, correctly. However, our method introduces the degree of context dependency in order to judge whether a word in a text is a true keyword or not. Therefore, our method can classify the test data into A or B, when the keyword of the category A is judged to be the word w_1. As a result, our method can represent the characteristic of the texts more precisely than *SVMV*.

Conclusion

We have reported on an empirical study for term weight learning for an automatic text categorization. The characteristic of our approach is that the degree of context dependency is introduced in order to judge whether a word in a text is a true keyword or not. In the experiment using *WSJ*, we could obtain 0.75 breakeven points for 4,453 texts which are classified into 78 categories.

In our current method, category assignment is based on a word in texts, i.e. every text which should be classified and texts which are pre-categorized are characterized by a vector, each dimension of which is associated with a word in texts. As a result, two words are treated quite different even if these words are semantically similar. In order to get more accuracy, linking words with their semantically similar words might be necessary to be introduced into our framework.

Acknowledgments

The authors would like to thank the reviewers for their valuable comments. This work was partially supported by the Grant-in-aid for Scientific Research of the Ministry of Education, Science and Culture of Japan (No. 09780322).

References

[Brill1992] E. Brill. 1992. A simple rule-based part of speech tagger. In *Proc. of the 3rd Conference on Applied Natural Language Processing*, pages 152–155.

[Fuhr1989] N. Fuhr. 1989. Models for retrieval with probabilistic indexing. *Information Processing & Retrieval*, 25(1):55–72.

[Fukumoto and Suzuki1996] F. Fukumoto and Y. Suzuki. 1996. An automatic clustering of articles using dictionary definitions. In *Proc. of the 16th International Conference on Computational Linguistics*, pages 406–411.

[Fukumoto et al.1997] F. Fukumoto, Y. Suzuki, and J. Fukumoto. 1997. An automatic extraction of key paragraphs based on context dependency. In *Proc. of the 5th Conference on Applied Natural Language Processing*, pages 291–298.

[Guthrie and Walker1994] L. Guthrie and E. Walker. 1994. Document classification by machine: Theory and practice. In *Proc. of the 15th International Conference on Computational Linguistics*, pages 1059–1063.

[Iwadera and Kikui1997] T. Iwadera and G. Kikui. 1997. Automatic text categorization using trend-tracking technique. In *Proc. of the Natural Language Processing Pacific Rim Symposium*, pages 645–648.

[Iwayama and Tokunaga1994] M. Iwayama and T. Tokunaga. 1994. A probabilistic model for text categorization: Based on a single random variable with multiple values. In *Proc. of the 4th Conference on Applied Natural Language Processing*, pages 162–167.

[Jardine and Sibson1968] N. Jardine and R. Sibson. 1968. The construction of hierarchic and non-hierarchic classifications. pages 177–184.

[Jones1973] K. S. Jones. 1973. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

[Kwok1989] K. L. Kwok. 1989. Experiments with a component theory of probabilistic information retrieval based on single terms as document components. *ACM Transactions on Information Systems*, 8(4):363–386.

[Lewis1992] D. Lewis. 1992. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR92*, pages 37–50.

[Liberman1991] M. Liberman, 1991. *CD-ROM I.* Association for Computational Linguistics Data Collection Initiative University of Pennsylvania.

[Luhn1958] H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM journal*, 2(1):159–165.

[Robertson and Jones1976] S. E. Robertson and K. Sparck Jones. 1976. Relevance weighting of search terms. Number 27, pages 129–146.

[Salton and Yang1973] G. Salton and C. S. Yang. 1973. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372.

[Salton1988] G. Salton. 1988. In *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley.

[Tokunaga and Iwayama1994] T. Tokunaga and M. Iwayama. 1994. Text categorization based on weighted inverse document frequency. *SIG-IPS Japan*, 100(5):33–40.

[Walker and Amsler1986] D. Walker and R. Amsler. 1986. In *The Use of Machine-Readable Dictionaries in Sublanguage Analysis*, pages 69–84. Lawrence Erlbaum, Hillsdale, NJ.

[Yuasa et al.1995] N. Yuasa, T. Ueda, and F. Togawa. 1995. Classifying articles using lexical co-occurrence in large document databases. *Trans. of Information Processing Society Japan (In Japanese)*, 36(8):1819–1827.