On a New Model for Automatic Text Categorization Based on Vector Space Model

Makoto Suzuki*, Naohide Yamagishi*, Takashi Ishida†, Masayuki Goto† and Shigeichi Hirasawa‡
*Faculty of Information Science, Shonan Institute of Technology
1-1-25 Tsujido Nishikaigan, Fujisawa, Kanagawa, Japan, 251-8511
Email: m-suzuki@info.shonan-it.ac.jp
†Waseda University, 3-4-1 Okubo Shinjuku-ku, Tokyo, Japan, 169-8555
‡Cyber University, 4F 1-11 Kitayamabushi-cho, Shinjuku-ku, Tokyo, Japan, 162-0853

***Abstract*—In our previous paper, we proposed a new classification technique called the Frequency Ratio Accumulation Method (FRAM). This is a simple technique that adds up the ratios of term frequencies among categories, and it is able to use index terms without limit. Then, we adopted the Character N-gram to form index terms, thereby improving FRAM. However, FRAM did not have a satisfactory mathematical basis. Therefore, we present here a new mathematical model based on a "Vector Space Model" and consider its implications. The proposed method is evaluated by performing several experiments. In these experiments, we classify newspaper articles from the English Reuters-21578 data set, a Japanese CD-Mainichi 2002 data set using the proposed method. The Reuters-21578 data set is a benchmark data set for automatic text categorization. It is shown that FRAM has good classification accuracy. Specifically, the micro-averaged F-measure of the proposed method is 92.2% for English. The proposed method can perform classification utilizing a single program and it is language-independent.**

***Index Terms*—text mining, classification, *N*-gram, newspaper**

I. Introduction

The spread of computers has rapidly increased the amount of accumulated electronic text. Recently, automatic text categorization has received a great deal of attention because it is becoming hard to manually classify enormous amounts of text for the purpose of, for example, category-based retrieval. This paper discusses automatic text categorization, which is the process of categorizing a document appropriately, using a pre-defined set of categories[1].

In general, automatic text categorization involves two important phases. The first, or training, phase, is the extraction of index terms to yield effective keywords for the second, or test, phase, which is the actual classification of documents using the index terms from the training phase. In the present paper, we refer to an important stage in the training phase as "feature selection" and that in the test phase as the "document classification" stage.

A single word is usually considered to be an index term in the feature selection stage. In a language with words delimited by spaces, such as English, there is no need to extract words. However, for Japanese, words should be extracted by morphological analysis. In contrast, a method to generate these index terms using a Character N-gram has been proposed as a language-independent technique[2],[3]. In any case, most of these conventional techniques extract useful index terms from many words by using mutual information, TFIDF values etc.[4], and these index terms are then used for classification.

On the other hand, categorization at the document classification stage is a traditional problem of machine learning, and machine learning algorithms are often used, such as Neural Networks[5], Decision Trees[6],[7], the Naive Bayes Method[8], k-Nearest Neighbor[9] and Boosting Algorithms[10], as well as Support Vector Machines (SVM)[11].

In our previous paper, we proposed a new classification technique called the Frequency Ratio Accumulation Method (FRAM)[12]. This is a simple technique that adds up the ratios of term frequency among categories, and is such that it can make use of index terms without limit. Then, we adopted a Character N-gram to form index terms, thereby improving FRAM, which performed well in comparison with other techniques. However, FRAM lacked a sound mathematical basis. Therefore, here we describe a new mathematical model based on the "Vector Space Model" and consider its implications. Furthermore, the proposed method is evaluated by performing several experiments. In these experiments, we classify newspaper articles from the English Reuters-21578 data set [1] , a Japanese CD-Mainichi 2002 data set [2] using the proposed method. In particular, we observe that the Reuters-21578 data set is a benchmark data set for automatic text categorization. As a result, we show that FRAM has good classification accuracy [3] More precisely, the micro-averaged F-measure of the proposed method is 92.2% for English. The proposed method can perform classification using a single program and it is language-independent.

II. Text Categorization

A. Overview

In this study, the goal of text categorization is to classify some given new documents into a fixed number of pre-defined categories. Figure 1 depicts a flow diagram for the text categorization task[13].

[1] http://www.daviddlewis.com/resources/testcollections/ reuters21578/

[2] CD-Mainichi Newspapers 2002 data, Nichigai Associates, Inc., 2003 (Japanese).

[3] We use the term "classification accuracy" here without distinguishing Precision, Recall and F-measure.

978-1-4244-6587-3/10/$25.00 É2010 IEEE

The procedure for automatic text categorization is divided into two phases, the training phase and the test phase. In the training phase, training documents are input, along with a category for each. Next, the index terms are extracted via a feature selection stage and an indices database is produced, referred to herein as the "index term set (DB)", which is later used for the test phase. In the test phase, new documents to be classified are input one after another, and a category is allocated to each new document with a classifier that uses methods such as the Naive Bayes Method, SVM, or our proposed method. Finally, the classification results of each technique are evaluated.

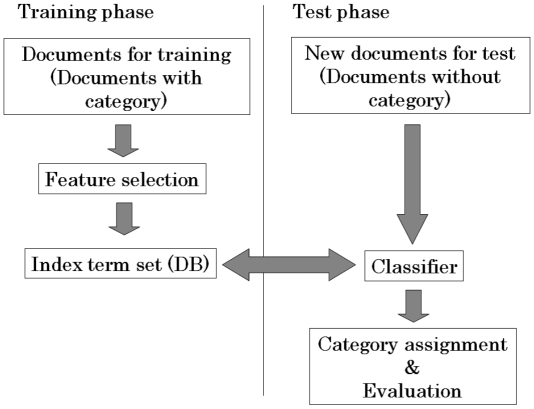

Training phase
Test phase
Documents for training (Documents with category)
New documents for test (Documents without category)
Feature selection
Index term set (DB)
Classifier
Category assignment & Evaluation

Fig. 1: Flow Diagram for Text Categorization

B. N-gram

We will use N-grams to generate index terms in the present paper. N-grams come in two variants, namely the "Character N-gram" and the "Word N-gram". An example for English is shown in Table I and an example for Japanese is shown in Table II [4].

TABLE I: Example of an N-gram in English

Original sentence	He is fine.
Word	He/is/fine
Character N-gram (N=2)	He/e / i/is/s / f/fi/in/ne
Word N-gram (N=2)	He is/is fine

TABLE II: Example of an N-gram in Japanese

Original sentence	彼は元気です.
Word	彼/は/元気/です
Character N-gram (N=2)	彼は/は元/元気/気で/です
Word N-gram (N=2)	彼は/は元気/元気です

As shown in Table I-II, a Character N-gram is a character string of N characters extracted from a sentence. We extract many character strings of N characters from the beginning of a sentence by moving along the sentence one character at a time. In contrast, a Word N-gram is an N-gram composed of N words. The Character N-gram is important in the present paper. In addition, the Character N-gram is effective as a language-independent method because it does not depend on the meaning of the language.

C. Mathematical Formulation

In the present paper, the following notations are used. First, the document vector for training is defined as follows.

Definition 1 (Training-Document-Vector):

$$\mathbf{d}_j = (d_{1j}, d_{2j}, \cdots, d_{Ij}) \tag{1}$$

d_{ij} : feature number [5] of term t_i contained in the j-th document

□

Next, the set of training-document-vectors is defined as follows.

Definition 2 (Document Set for Training):

$$D = \{\mathbf{d}_j \mid j = 1, 2, \cdots, J\} \tag{2}$$

$\mathbf{d}_j$: the j-th document for learning
J : total number of documents for learning

□

We use "term" without distinguishing an N-gram from a word here[6].

Definition 3 (Term Set):

$$T = \{t_i \mid i = 1, 2, \cdots, I\} \tag{3}$$

I : total number of terms contained in all documents

□

Moreover, the new document that is the object of classification is defined as follows.

Definition 4 (New Document Vector):

$$\mathbf{d} = (d_1, d_2, \cdots, d_I) \tag{4}$$

d_i : feature number of term t_i contained in the new document

□

In addition, the set of categories to which each document belongs is written as follows.

Definition 5 (Category Set):

$$C = \{c_k \mid k = 1, 2, \cdots, K\} \tag{5}$$

c_k : a category
K : total number of categories

□

Finally, the data for learning are given as follows. The learning data is a pair, consisting of a document vector and its category or categories [7].

[4] These sentences carry the same meaning.

[5] The d_{ij} in Eq(1) is the feature number. For example, the $tfidf_{ij}$ in Eq(9) is equivalent to d_{ij}. In contrast, the e_{ij}. in Eq(16) is equivalent to d_{ij} in Eq(1). Moreover, each document is $\mathbf{d}_{\mathbf{f}j} \in \mathscr{R}$ in Eq(9) whereas it is $\mathbf{d}_{\mathbf{e}j} \in \mathscr{Z}^I$ in Eq(16). Here, $\mathscr{R}$ denotes the set of real numbers and $\mathscr{Z}$ denotes the integers.

[6] We referred to such as an "index term" above.

[7] Each document in the Japanese CD-Mainichi 2002 data set belongs to a single category. On the other hand, the English Reuters-21578 data set includes documents that belong to several categories.

Definition 6 (Data for Training):

$$(D, L) = \{\ (\mathbf{d}_j,\ \ \mathbf{l}_j) \mid j = 1, 2, \cdots, J\ \} \tag{6}$$

where

$$\mathbf{l}_j = (l_{j1}, l_{j2}, \cdots, l_{jK})$$

$$l_{jk} = \begin{cases} 1 & if\ \ \mathbf{d}_j \in D(c_k) \\ 0 & if\ \ \mathbf{d}_j \notin D(c_k) \end{cases}$$

$\mathbf{l}_j$: K-dimensional category label vector that represents the categories of a training document $\mathbf{d}_j$ $(j = 1, 2, \cdots, J)$
L : a set of category label vectors $\mathbf{l}_j$
$D(c_k)$: a set of training document $\mathbf{d}_j$ that belongs to the category c_k □

For example, when the total number of categories is 3 $(K = 3)$, $\mathbf{l}_j = (1, 0, 0)$ if $\mathbf{d}_j$ belongs to a single category c_1. In contrast, $\mathbf{l}_j = (1, 1, 0)$ if $\mathbf{d}_j$ belongs to two categories c_1 and c_2.

Using these notations, the problem of automatic text categorization in the present paper is to classify a new document $\mathbf{d}$ into a pre-defined unknown category $\hat{c}$, given a set of training data (D, L) and $\mathbf{d}$.

III. Previous Method - Vector Space Model -

As mentioned in Section I, many machine learning algorithms have been used for document classification so far. The survey[1] describes the performance of these techniques in detail. In addition, some probabilistic models based on Bayesian Statistics have been proposed[14]. However, these techniques are not suitable here because of the greatly increased computational complexity when a large number of documents are to be classified. Here, we will explain the "Vector Space Model (VSM)" that is the basis for our method.

A. Mathematical Formulation of the Vector Space Model

In VSM, the following TFIDF value is often used in place of the total number of terms, that is d_{ij} in Eq(1) and d_i in Eq(4), as an element of a training-document-vector and a new document vector.

Definition 7 (TFIDF Value):

$$tfidf_i = tf(t_i, \mathbf{d}) \cdot idf(t_i) \tag{7}$$

where

$$idf(t_i) = \log \frac{J}{df(t_i)} + 1 \tag{8}$$

J : total number of documents for learning
$tf(t_i, \mathbf{d})$: total number of terms t_i contained in a document $\mathbf{d}$
$df(t_i)$: the number of documents containing the term t_i □

If we use the TFIDF value, each training-document-vector $\mathbf{d_f}_j$ and a new document vector $\mathbf{d}_\eta$ can be expressed as follows [8].

Definition 8 (Training-Document-Vector):

$$\mathbf{d_f}_j = (tfidf_{1j}, tfidf_{2j}, \cdots, tfidf_{Ij}) \in \mathscr{R} \tag{9}$$

$\mathscr{R}$: I-dimensional vector of real numbers □

Definition 9 (New Document Vector):

$$\mathbf{d}_\eta = (tfidf_1, tfidf_2, \cdots, tfidf_I) \in \mathscr{R} \tag{10}$$

□

In addition, we define the center of gravity vector of all training-document-vectors that belong to each category c_k. We call this vector a "Category-Representative-Vector" in the present paper.

Definition 10 (Category-Representative-Vector):

$$\begin{aligned} \mathbf{r}_k &= \frac{1}{N_{c_k}} \sum_{\mathbf{d_f}_j \in D(c_k)} \mathbf{d_f}_j \\ &= (r_{1k}, r_{2k}, \cdots, r_{Ik}) \in \mathscr{R} \end{aligned} \tag{11}$$

N_{c_k} : total number of all training-document-vectors that belong to each category c_k
$D(c_k)$: a set of training document $\mathbf{d}_j$ that belongs to the category c_k □

Finally, we define a measure of similarity as the cosine between the "Category-Representative-Vector" and the "New Document Vector" in VSM.

Definition 11 (Similarity in the Vector Space Model):

$$sim_V(\mathbf{r}_k, \mathbf{d}_\eta) = \frac{\langle \mathbf{r}_k, \mathbf{d}_\eta \rangle}{\mid \mathbf{r}_k \mid\mid \mathbf{d}_\eta \mid} \tag{12}$$

$\mathbf{r}_k$: category-representative-vector in the Vector Space Model
$\mathbf{d}_\eta$: new document vector that is the object of classification □

B. Procedure of Categorization Based on VSM

Using the mathematical formulation developed above, we describe a standard procedure for automatic classification based on VSM as follows.

1. Extract many terms from documents for learning.
2. Count the frequency of each term in every document.
3. Extract those terms to be used for the classification if necessary [9].
4. Represent each document as a document vector composed of terms which are chosen in Step3 using TFIDF values in Eq(7). Here, the document vector is given by Eq(9).
5. Calculate the "Category-Representative-Vector" of each category via Eq(11) using the training-document-vectors in Step 4.
6. Represent the new document which is an object of classification as a document vector as in Eq(10). Next, calculate the similarity between the new document vector and each category-representative-vector using Eq(12).

[8] In practice, we tend to use normalized TFIDF values, that are divided by the length of the document, in place of TFIDF in Eq(9).

[9] This is "Feature Selection" in Figure 1. For example, we can use "mutual information" as the criterion for feature selection.

Finally, compare the new document with every category-representative-vector, and classify it as belonging to a category with the highest similarity, as determined by Eq(13).

$$\hat{c} \quad = \quad \arg\max_{c_k \in C} sim_V(\mathbf{r}_k, \mathbf{d}_\eta) \tag{13}$$

In this way, we finally classify the new document $\mathbf{d}_\eta$ as belonging to the category $\hat{c}$ with highest similarity in VSM.

IV. Proposed Method

Here, we will explain the "Accumulation Method" that is proposed in the present paper, considering two versions, AM1 and AM2.

A. Proposed Method AM1, with Experiments

1) Mathematical Formulation of the Accumulation Method: An important problem in automatic document classification concerns how we process a three-dimensional sparse matrix, used for denoting terms, documents, and categories.

$$A = \left[\ e_{ijk}\ \right] \in \mathscr{X}^{I \times J \times K} \tag{14}$$

$$e_{ijk} = \begin{cases} 1 & if \quad \mathbf{d}_j \in D(c_k) \text{ and } tf(t_i, \mathbf{d}_j) \geq 1 \\ 0 & \text{except the above} \end{cases}$$

In VSM, we usually obtain a three-dimensional matrix by taking into account terms and documents, we can obtain the term-category-matrix shown in Definition 12 that is a projection of the three-dimensional matrix above to the subspace of terms and categories for our Accumulation Method (AM).

Definition 12 (Term-Category-Matrix):

$$B = \left[\ e_{i \cdot k}\ \right] \in \mathscr{X}^{I \times K} \tag{15}$$

$e_{i \cdot k}$: total number of documents that include term t_i and belong to the k-th category
$\mathscr{X}^{I \times K}$: $I \times K$ integer matrix space

□

We now consider how to represent documents for training in the proposed method. We represented training-document-vectors in Eq(9) using TFIDF values in VSM, whereas we use the binary vector shown in Eq(16) in the proposed method. In other words, this document vector $\mathbf{d}_{\mathbf{e}j}$ is a binary vector such that the i-th element is 1 if there is a term t_i in the training-document $\mathbf{d}_{\mathbf{e}j}$.

Definition 13 (Training-Document-Vector):

$$\mathbf{d}_{\mathbf{e}j} = (e_{1j\cdot}, e_{2j\cdot}, \cdots, e_{Ij\cdot}) \tag{16}$$

where

$$e_{ij\cdot} = \begin{cases} 1 & if \quad tf(t_i, \mathbf{d}_{\mathbf{e}j}) \geq 1 \\ 0 & if \quad tf(t_i, \mathbf{d}_{\mathbf{e}j}) = 0 \end{cases}$$

$tf(t_i, \mathbf{d}_{\mathbf{e}j})$: total number of terms t_i contained in a document $\mathbf{d}_{\mathbf{e}j}$

□

In the same way, we will represent a new document $\mathbf{d}_\epsilon$ that is the object of classification.

Definition 14 (New Document Vector):

$$\mathbf{d}_\epsilon = (\epsilon_1, \epsilon_2, \cdots, \epsilon_I) \tag{17}$$

where

$$\epsilon_i = \begin{cases} 1 & if \quad tf(t_i, \mathbf{d}_\epsilon) \geq 1 \\ 0 & if \quad tf(t_i, \mathbf{d}_\epsilon) = 0 \end{cases}$$

□

In addition, we will define p_{ik} as follows. We can regard p_{ik} as the conditional probability $P(c_k|t_i)$ that the term t_i belongs to category c_k when t_i appears.

Definition 15 (Category-Representative-Vector):

$$\tilde{\mathbf{r}_{\mathbf{p}}}_k = (p_{1k}, p_{2k}, \cdots, p_{Ik}) \in \mathscr{R} \tag{18}$$

where

$$p_{ik} = \frac{e_{i \cdot k}}{\sum_{k=1}^{K} e_{i \cdot k}} \equiv P(c_k|t_i) \tag{19}$$

$\mathscr{R}$: I-dimensional vector of real numbers

□

Finally, we will define similarity for AM. This is an inner product, and it is defferent from the similarity as defined in Definition 11.

Definition 16 (Similarity in AM):

$$sim_A(\tilde{\mathbf{r}_{\mathbf{p}}}_k, \mathbf{d}_\epsilon) = \langle \tilde{\mathbf{r}_{\mathbf{p}}}_k, \mathbf{d}_\epsilon \rangle \tag{20}$$

$\tilde{\mathbf{r}_{\mathbf{p}}}_k$: category-representative-vector of category c_k in AM1
$\mathbf{d}_\epsilon$: a new document vector that is the object of classification

□

2) Procedure of Categorization Based on AM:

1. Extract many terms from documents for learning.
2. Identify the existence of each term in every document.
3. Extract those terms to be used for classification if necessary.
4. Generate a term-category-matrix as shown in Eq(15), using the number of each document for learning.
5. Calculate each category-representative-vector as shown in Eq(18) using the conditional probabilities in Eq(19).
6. Represent the new document which is an object of classification as a document vector as given by Eq(17). Next, calculate the similarity between the new document vector and each category-representative-vector using Eq(20). Finally, compare the new document with every category-representative-vector, and classify it into a category with the highest similarity, as given by Eq(21).

$$\hat{c} \quad = \quad \arg\max_{c_k \in C} sim_A(\tilde{\mathbf{r}_{\mathbf{p}}}_k, \mathbf{d}_\epsilon) \tag{21}$$

The basic procedure is similar to the classification in VSM discussed in Section III-B. However, there are some differences in Steps 2-6, which we expand upon here.

Firstly, we identify only the existence of each term in every document in Step 2, instead of counting its frequency. Thus we use only binary vectors in Eq(16). Secondly, we do not extract features (index terms) in Step 3. Of course, classification accuracy can be improved by choosing features in the proposed method[15]. However, we used all terms that appeared in training-documents to omit unnecessary operations in the present study. Thirdly, we use only the number of documents including the term without using special measures, such as TFIDF values, in Step 4. Fourthly, we calculate each category-representative-vector in Step 5 using only the number of documents maintained by a term-category-matrix in the proposed method. In addition, the numerical values used in this case are the ratios mentioned in Eq(19), thus they are not TFIDF values. On the other hand, VSM represents each training-document as a document vector and calculates category-representative-vectors with TFIDF values as shown in Steps 4-5 of Section III-B. In other words, we do not have to store the above information when using the Accumulation Method and have only to maintain the $I \times K$ term-category-matrix [10] at the end of the training phase. Fifthly, we use a binary vector in Eq(17) for the new document vector in Step 6, too. In other words, we maintain only binary values that specify whether each term is included in the document, and do not have to maintain the frequency of each term included in it. Moreover, the calculation of the similarity is different. That is to say, VSM uses the cosine in Eq(12), whereas AM uses the inner product in Eq(20).

3) Experiment and Results with the Proposed Method:

(3-1) Experimental Conditions

The present experiment involved two newspapers that contained articles with pre-assigned categories. The first is the English Reuters-21578 data set, the second is the Japanese CD-Mainichi 2002 data set.

Here, the English Reuters-21578 data set provides benchmark data for automatic text categorization. An *Apte split with 10 categories* was used for Reuters-21578. The *Apte split with 10 categories* is benchmark data that has extracted ten categories, named *Acquisition*, *Corn*, *Crude*, *Earn*, *Grain*, *Interest*, *Money-fx*, *Ship*, *Trade*, and *Wheat* from Reuters-21578.

In addition, the Japanese CD-Mainichi 2002 data has extracted from it seven categories, named *Society*, *Sports*, *Entertainment*, *Home*, *Economy*, *International Relations*, and *Leaders*. We randomly selected 1,000 training documents and 500 test documents (7,000 and 3,500 documents in total, respectively) for each category.

We classified the two types of newspaper articles mentioned above using six methods for the English Reuters-21578 data and the Japanese CD-Mainichi 2002 data as shown in Table III. In the present experiment, for each method, the computer was initially made to learn using training data with pre-assigned categories in the training phase. Secondly, in the test phase, we gave the test data to the computer without showing them their true categories, and made the computer classify them.

(3-2) Measuring Classification Performance

Precision, Recall and F-measure are the most commonly used measures for evaluating text categorization or information retrieval systems. These measures are calculated based on the following table IV.

Here, TP_{c_k}, FP_{c_k}, FN_{c_k}, and TN_{c_k} are counts that reflect how the assigned categories matched the correct categories.

TABLE III: Each Method in the Present Experiment

Ex.-No.	Reuters (in English)	Mainichi (in Japanese)
1	Character 8-gram	Character 3-gram
2	Character 9-gram	Character 4-gram
3	Character 10-gram	Character 5-gram
4	Character 11-gram	Character 6-gram
5	Character 12-gram	Character 7-gram
6	Character 13-gram	Character 8-gram

TABLE IV: Example of Judgement about Category C_k

Category C_k		**Newspaper Assigned (Correct)**	
		TRUE	**FALSE**
Classifier Assigned (Judgement)	**TRUE**	TPc_k	FPc_k
	FALSE	FNc_k	TNc_k

Using these frequencies, Precision and Recall can be defined as follows.

Definition 17 (Precision):

$$P_{c_k} = \frac{TP_{c_k}}{TP_{c_k} + FP_{c_k}} \tag{22}$$

□

Definition 18 (Recall):

$$R_{c_k} = \frac{TP_{c_k}}{TP_{c_k} + FN_{c_k}} \tag{23}$$

□

Here, Recall is a measure to denote the ratio of the number of documents which the classifier was able to classify correctly to the total number of documents which it tried, and Precision is a measure to denote the ratio of the number of documents which it was able to classify correctly to the number of documents contained in each category which it used for classification. Both measures can be calculated for each category and then averaged, or can be calculated over all decisions and then averaged. The former is called "macro-averaging", and the latter is called "micro-averaging". We arrange these and depict them in Figure 2.

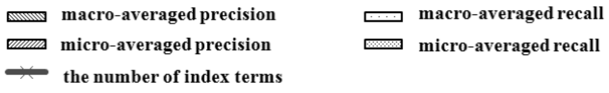

▨ macro-averaged precision
▨ micro-averaged precision
the number of index terms
macro-averaged recall
▨ micro-averaged recall

Fig. 2: Legend for Figures 3 - 6

There is also the "F-measure", which is the harmonic mean between precision and recall. It is defined as follows.

[10] Its dimension is not greater than $I \times K$.

Definition 19 (F-measure):

$$F_{c_k} = \frac{2P_{c_k}R_{c_k}}{P_{c_k}+R_{c_k}} = \frac{2TP_{c_k}}{2TP_{c_k}+FP_{c_k}+FN_{c_k}} \tag{24}$$

□

In Figures 3 - 6, the left side value denotes the macro-averaged F-measure and the right side value denotes the micro-averaged F-measure of each bar in these graphs. These measures fall in the range from 0 to 1, with 1 being the best score.

(3-3) Results

These results are shown in Figures 3 - 4. Figure 3 shows results for the case of the English Reuters-21578 data and Figure 4 shows results for the case of the Japanese CD-Mainichi 2002 data. Figure 3 shows that the highest micro-averaged F-measure is 92.0% in case of N=11 for English, and Figure 4 shows that the best value is 88.2% in case of N=6 for Japanese. When the number of training-documents is equal in each category, such as for Japanese as shown in Figures 4, the proposed method AM1 shows good performance for which the macro-average and the micro-average are almost the same. However, when the number of training-documents in each category is not uniform, such as for the Reuters data set, as shown in Figure 3, the method AM1 produces a much inferior macro-averaged precision and recall compared with the micro-averaged precision and recall. Especially, we can not calculate the macro-averaged precision because there are two categories, *Corn* and *Wheat*, where no document is allocated.

B. Proposed Method 2 and Its Experiments

In this section, we propose a second method that improves on the method AM1. Then, we will discuss the experiments and results using the method AM2.

1) Improvement of the Proposed Method:

As shown in Definition 15, we calculated each category-representative-vector via Eq(18) using the conditional probability p_{ik} in Eq(19) in the proposed method AM1. The results are shown in Figures 3 - 4. However, the method AM1 produces a much inferior macro-averaged precision and recall compared with the micro-averaged precision and recall as mentioned the above. Therefore, we consider the case where the number of learning documents is not uniform among categories and use the "Revised-Category-Representative-Vector" in Definition 20. In other words, we use q_{ik} in Eq(26) not p_{ik} in Eq(19), i.e., we divided by the sum over p_{ik} for each category.

Definition 20 (Revised-Category-Representative-Vector):

$$\mathbf{r}_{\tilde{\mathbf{q}}_k} = (q_{1k}, q_{2k}, \cdots, q_{Ik}) \in \mathscr{R} \tag{25}$$

where,

$$q_{ik} = \frac{p_{ik}}{p_{\cdot k}} = \frac{p_{ik}}{\sum\limits_{i=1}^{I} p_{ik}} \tag{26}$$

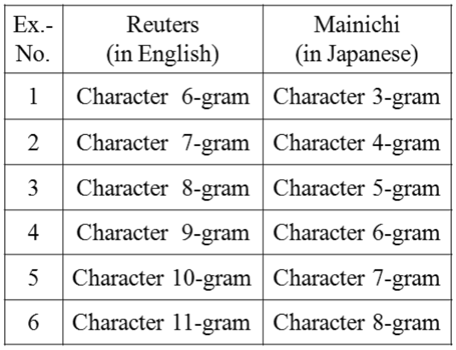
□

The procedure except for $\mathbf{r}_{\tilde{\mathbf{q}}_k}$ in Eq(25) of the proposed method AM2 is the same as for the method AM1, and the calculation of similarity has only to replace $\mathbf{r}_{\tilde{\mathbf{p}}_k}$ in Eq(20) with $\mathbf{r}_{\tilde{\mathbf{q}}_k}$ in Eq(25).

2) Experiment and Result for the Proposed Method AM2:

We will show the conditions for the experiments with the proposed method AM2 in Table V. Table V is the same as Table III for Mainichi. However, the values of N for the Character N-gram become slightly small compared with Table III for the Reuters data.

TABLE V: Each Method in the Experiment 2 with Method AM2



Ex.- No.	Reuters (in English)	Mainichi (in Japanese)
1	Character 6-gram	Character 3-gram
2	Character 7-gram	Character 4-gram
3	Character 8-gram	Character 5-gram
4	Character 9-gram	Character 6-gram
5	Character 10-gram	Character 7-gram
6	Character 11-gram	Character 8-gram

We will write results for the experiment with the proposed method AM2 as follows, because other conditions including the measure of evaluation are the same as for method AM1 detailed in Section IV-A3.

As shown in Figures 5 - 6, the highest micro-averaged F-measure with the proposed method is 92.2% in case of N=9 for English Reuters-21578 data. In contrast, it is 88.9% in case of N=5 for the Japanese CD-Mainichi 2002 data.

According to the comparison table [11] of the reference[1], the classification accuracy of a support vector machine that had generated the most precise classification was 92.0%. This suggests that the proposed method AM2 can generate a sophisticated classification compared with other techniques.

V. Discussion

With the calculation of the similarity in Eq(20) for the proposed method, it is considered that the addition of the possibility that a new document $\mathbf{d}$ belongs to each category c_k when a term t_i included in the document $\mathbf{d}$ appeared in the category c_k. Here, many of the probabilities in Eq(19) are zero or are very small values close to zero. In general, it is known that the following approximate expression applies in the case that the probabilities p_i are very small, where p_i are the occurrence probabilities of mutually independent events E_i.

$$1 - \prod_i (1-p_i) \simeq \sum_i p_i \tag{27}$$

If we assume that p_i is the conditional probability $P(c_k|t_i)$ in Definition 15, the calculation of similarity using the proposed method corresponds to the right hand side of Eq(27).

[11] In the present paper, we used the data ♯5 of p.38 in reference[1].

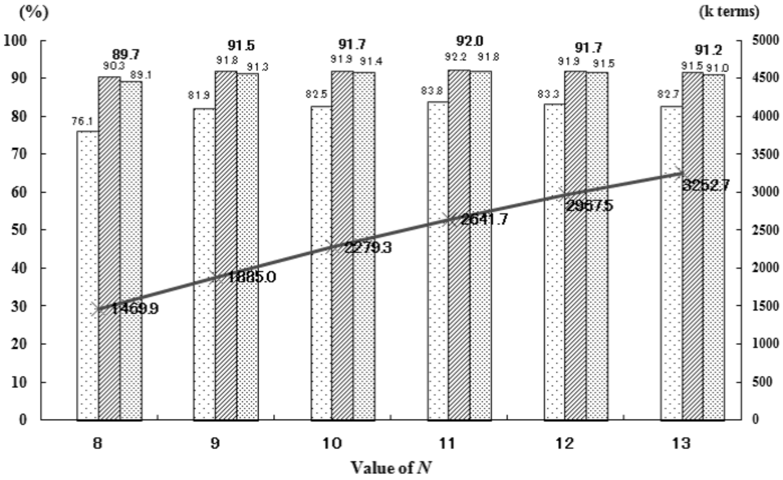


Fig. 3: Results of Proposed Method AM1 for English

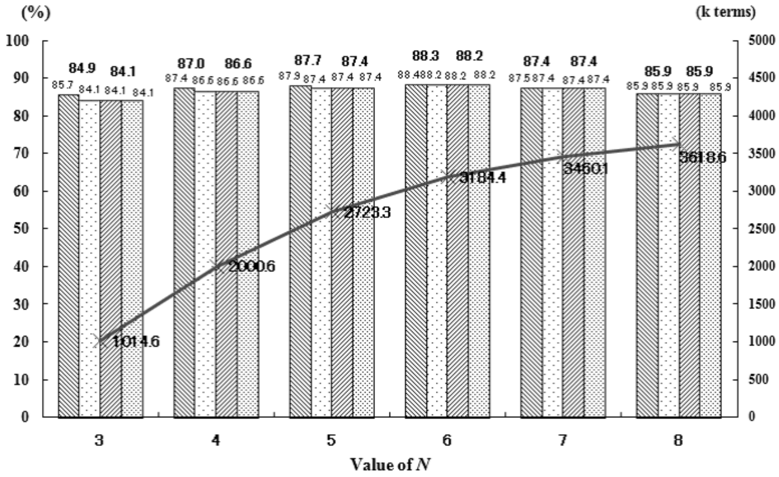


Fig. 4: Results of Proposed Method AM1 for Japanese

Therefore, we can understand that the value calculated by this addition is the probability of the union of events E_i, as shown on the left hand side of Eq(27). For example, we will calculate the probability that a new document $\mathbf{d}$ belongs to a category c_1 named "International Relations" by this addition if a term t_1 named "Obama" and a term t_2 named "Barack" were included in the document $\mathbf{d}$. In previous techniques based on probabilistic models, the possibility of co-occurrence terms is usually calculated by multiplication, considered as the probability of the intersection. On the other hand, we calculated using only addition, by considering a union. In other words, if a document contains co-occurrence terms, the probability that the document belongs to the category becomes closer to 0 using previous techniques, because two small probabilities, which are close to zero, are multiplied. However, the probability that the document belongs to the category increases using the proposed method when the document includes the co-occurrence terms. In this way, since we can deal with the particular problem that many of the probabilities $P(c_k|t_i)$ are zero or close to zero in tasks such as text categorization, we think that our new idea is important.

VI. Conclusion

In the present paper, we have proposed a new mathematical model of automatic text categorization and a classification method based on VSM. Moreover, we have shown that the proposed method has good classification accuracy by several experiments. These experiments used the English Reuters-21578 data set and the Japanese CD-Mainichi 2002 data set. Specifically, the micro-averaged F-measure of the proposed method is 92.2% for English. The Reuters-21578 data set is a benchmark newspaper article data set for automatic document classification. The proposed method can perform classification using a single program and it is language-independent.

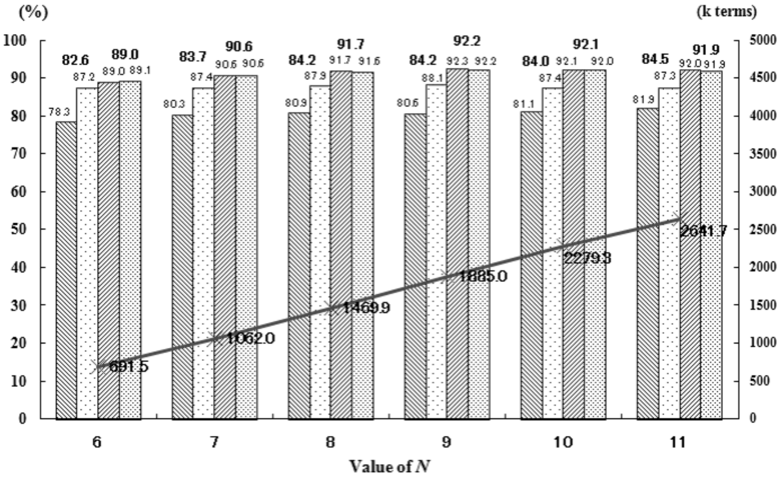


Fig. 5: Results of Proposed Method AM2 for English

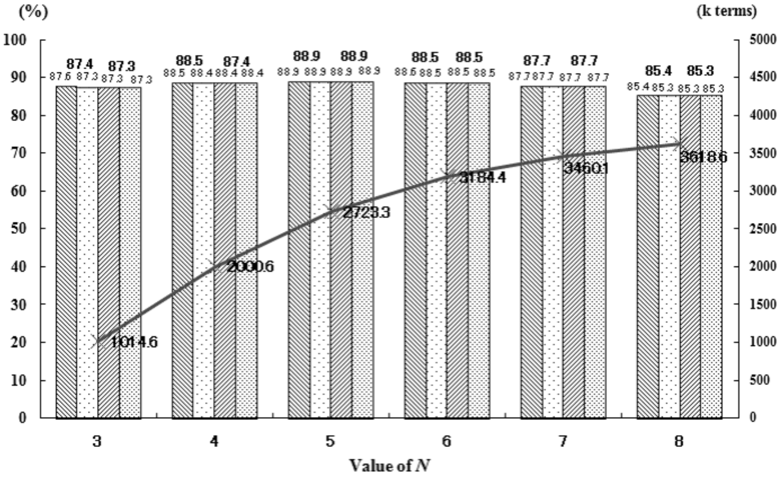


Fig. 6: Results of Proposed Method AM2 for Japanese

ACKNOWLEDGMENT

This work was supported by a Grant-in-Aid for Scientific Research (KAKENHI-C) 22500140.

REFERENCES

[1] F. Sebastiani: "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, Vol.34, pp.1–47, 2002.

[2] W. Cavnar and J. Trenkle: "N-Gram-Based Text Categorization," *Proc. 3rd Annual Sympo. on Document Analysis and Information Retrieval (SDAIR)*, pp.161–169, 1994.

[3] P. Nather: "N-gram based Text Categorization," *Diploma thesis, Comenius Univ., Faculty of Mathematics, Physics and Informatics, Institute of Informatics*, 2005.

[4] A. Aizawa: "The Index Quantity: An Information Theoretic Perspective of Tfidf-like Measures," *ACM Computing SurveysProc. 23th ACM Int. Conf. on Research and Development in Information Retrieval*, pp.104-111, 2000.

[5] E.D. Wiener, J.O. Pedersen, and A.S. Weigend: "A neural network approach to topic spotting," *Proc. 4th Sympo. on Document Analysis and Information Retrieval (SDAIR)*, pp.317–332, 1995.

[6] C. Apte, F. Damerau and S.M. Weiss: "Automated Learning of Decision Rules for Text Categorization," *ACM Trans. of Information Systems*, Vol.12, No.3, pp.223–251, 1994.

[7] R. Rastogi and K. Shim: "A decision tree classifier that integrates building and pruning," *Proc. 24th Int. Conf. on Very Large Data Bases*, pp.404–415, 1998.

[8] D.D. Lewis and M. Ringuette: "A comparison of two learning algorithms for text categorization," *Proc. 3rd Annual Sympo. on Document Analysis and Information Retrieval (SDAIR)*, pp.81–93, 1994.

[9] Y. Yang: "An Evaluation of Statistical Approaches to Text Categorization," *Journal of Information Retrieval*, Vol.1, No.1, pp.67–88, 1999.

[10] R.E. Schapire and Y. Singer: "BoosTexter - A Boosting-based System for Text Categorization," *Machine Learning*, Vol.39, No.2-3, pp.135–168, 2000.

[11] T. Joachims: "Text categorization with support vector machines: learning with many relevant features," *Proc. 10th European Conf. on Machine Learning*, No.1398, pp.137–142, 1998.

[12] M. Suzuki and S. Hirasawa: "Text categorization based on the ratio of word frequency in each category," *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, pp.3535–3540, 2007.

[13] S.M. Namburu, H. Tu, J. Luo and K.R. Pattipati: "Experiments on Supervised Learning Algorithms for Text Categorization," *Proc. IEEE Aerospace Conf., Big Sky, MT*, pp.1–8,2005.

[14] D.M. Blei, A.Y. Ng and M.I. Jordan: "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, Vol.3, pp.993–1022, 2003.

[15] M. Suzuki, T. Ishida and M. Goto: "Refinement of Index Term Set and Improvement of Classification Accuracy," *Proc. of Int. Sympo. on Information Theory and its Applications (ISITA2008)*, pp.449–454, 2008.