

Two Step POS Selection for SVM Based Text Categorization

Takeshi MASUYAMA[†], *Student Member and* Hiroshi NAKAGAWA[†], *Member*

SUMMARY

Although many researchers have verified the superiority of Support Vector Machine (SVM) on text categorization tasks, some recent papers have reported much lower performance of SVM based text categorization methods when focusing on all types of parts of speech (POS) as input words and treating large numbers of training documents. This was caused by the overfitting problem that SVM sometimes selected unsuitable support vectors for each category in the training set. To avoid the overfitting problem, we propose a two step text categorization method with a variable cascaded feature selection (VCFS) using SVM. VCFS method selects a pair of the best number of words and the best POS combination for each category at each step of the cascade. We made use of the difference of words with the highest mutual information for each category on each POS combination. Through the experiments, we confirmed the validation of VCFS method compared with other SVM based text categorization methods, since our results showed that the macro-averaged F_1 measure (64.8%) of VCFS method was significantly better than any reported F_1 measures, though the micro-averaged F_1 measure (85.4%) of VCFS method was similar to them.

key words:

Text Categorization, Text Classification, Support Vector Machine (SVM), Parts of Speech (POS), Variable Cascaded Feature Selection (VCFS)

1. Introduction

The number of electronic documents, such as newspaper articles and patent documents, has increased with the explosive use of the Internet and online databases. As the available electronic documents increase, the demand for high precision systems in real world applications becomes more and more apparent [16]. For example, when using a search engine, the users can only afford to read the top few documents retrieved for a query, and therefore a search engine with high precision would be preferred to one with a high recall but low precision. In text categorization systems, when a classifier is used to help users decide the categories relevant to a document, the users tend to read only a few documents from each category, and therefore a classifier with high precision would be preferred to one with a high recall but low precision. Thus, our purpose is to propose an automatic text categorization system with high precision, keeping at least the same F_1 measure described in Sect. 4.2.

Many researchers have so far applied many machine learning methods to automatic text categorization for helping users utilize large numbers of documents. The methods are represented by Naive Bayes, Rocchio, k -Nearest Neighbor, Boosting, and Support Vector Machine (SVM) [2],[7],[17]. We focused on SVM in this paper, since published papers for automatic text categorization have verified the superiority of SVM based methods over other text categorization methods especially when using Reuters-21578 corpus* [13].

A major difficulty in text categorization methods is that too many input words are to be used to categorize. For example, Fukumoto et al. reported much lower performance (28.5% by F_1 measure) when they applied SVM to Reuters Corpus Volume I (RCV1)** [5]. This was caused by the overfitting problem that SVM sometimes selected unsuitable support vectors for each category in the training set, since they focused on all types of POS and treated large numbers of training documents. Therefore, we used mutual information to reduce the number of input words and avoid the overfitting problem, since mutual information has been shown to yield good performance for feature selection of SVM based text categorization methods [2].

Taira et al. have investigated the validation of mutual information filtering and POS filtering applying SVM to Japanese newspaper articles***. They reported that though the best number of words with the highest mutual information and the best POS combination differed greatly among categories, it was difficult to predict them [14]. Therefore, we propose a two step text categorization method with a variable cascaded feature selection (VCFS) to predict a pair of the best number of words with the highest mutual information and the best POS combination for each category at each step of the cascade.

VCFS method consists of two steps. At step 1, SVM classifies test documents either in a positive or a negative set. At step 2, SVM again classifies the test documents which belong to the positive set of step 1. We focused on the difference of words with the highest mutual information for each category on each POS combination. About the category "groundnut"

Manuscript received May 30, 2003.

Manuscript revised September 1, 2003.

[†]The authors are with Information Technology Center, The University of Tokyo, Tokyo, 113-0033 Japan.

*<http://www.daviddlewis.com/>

**<http://about.reuters.com/researchandstandards/corpus/>

***They used Mainichi Shinbun (Japanese newspaper articles) published in 1994 [11].

of Reuters-21578 corpus, for example, “oil,” “crude,” and “petroleum” are selected for the highest mutual information when our system selects words only from nouns, verbs, adjectives, and adverbs. On the other hand, these words are not selected due to their low mutual information when our system selects words from all types of POS. This indicates that selecting words only from nouns, verbs, adjectives, and adverbs is more powerful than selecting words from all types of POS to categorize test documents into the category “groundnut.”

The rest of this paper is organized as follows. Section 2 describes a basic framework of SVM. In Sect. 3, we present our categorization method. Section 4 shows some experimental results using Reuters-21578 corpus followed by evaluation and discussion. In Sect. 5, we describe conclusions.

2. Support Vector Machine

SVM [15] is a machine learning method for solving two-class pattern recognition problems. About natural language processing (NLP) research, many NLP researchers have applied SVM to a variety of problems, such as morphology and summarization, not to mention text categorization.

SVM learns from a training set to find a decision surface (classifier) which separates a set of positive examples (documents) from a set of negative examples by introducing the maximum margin between the two sets. The training set can be described by l points in the n -dimensional space $\mathbf{x}_i \in \mathbf{R}^n$ with two different labels $y_i \in \{-1, +1\}$ depending on the class which is assigned to the point \mathbf{x}_i for all $i = 1, \dots, l$.

Since the training set is chosen as linearly separable, there will be a hyperplane, which will be able to separate positive examples from negative examples. The points \mathbf{x}_i , which lie on the hyperplane, satisfy $\mathbf{w} \cdot \mathbf{x}_i + b = 0$, where $\mathbf{w} \in \mathbf{R}^n$ is the normal vector of the hyperplane, $\frac{|-b|}{\|\mathbf{w}\|}$ is the perpendicular distance from the hyperplane to the origin, and $\|\mathbf{w}\|$ is the Euclidean norm of \mathbf{w} . The variable $d(+)$ denotes the shortest distance from the separating hyperplane to the closest positive examples which satisfy a hyperplane H1 : $\mathbf{w} \cdot \mathbf{x}_i + b = 1$. Similarly, $d(-)$ denotes the shortest distance between the separating hyperplane and the closest negative examples which satisfy a hyperplane H2 : $\mathbf{w} \cdot \mathbf{x}_i + b = -1$. The distance between H1 and H2 is called “margin.” The margin can be calculated as $d(+)+d(-)$.

Mathematically, the points \mathbf{x}_i can be expressed by two inequalities as follows.

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \text{ for } y_i = +1 \quad (1)$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \text{ for } y_i = -1 \quad (2)$$

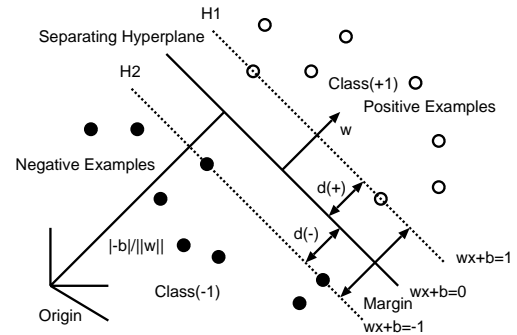


Fig. 1 Separating hyperplane.

The distance from H1 to the origin can be calculated as $\frac{|-b|}{\|\mathbf{w}\|}$. Similarly, the distance from H2 to the origin can be calculated as $\frac{|-1-b|}{\|\mathbf{w}\|}$. Comparing the two positions of H1 and H2 one can conclude $d(+)=d(-)=\frac{1}{\|\mathbf{w}\|}$. Furthermore, the margin, which reflects the distance between the two hyperplanes H1 and H2, equals $\frac{2}{\|\mathbf{w}\|}$. To maximize the margin, we should minimize $\|\mathbf{w}\|$. Fig. 1 depicts the situation for the linearly separable case in 2 dimensions.

The training documents which lie on either H1 or H2 are called “support vectors.” It is known that only the support vectors are used for categorization. This indicates that SVM causes low performance if SVM selects unsuitable support vectors which don’t represent the characteristics of a category [9].

Although we have so far focused on linear hypotheses, SVM can handle non-linear hypotheses by introducing a kernel function such as polynomial kernel, RBF kernel, and sigmoid kernel.

Since SVM is a binary classifier, we have to extend it to a multi-class classifier which classifies documents in three or more categories. Although there are many approaches to extend SVM to a multi-class classifier, we introduce two well known approaches. One is “one-against-all” approach which constructs k classifiers, one for each category. The k -th classifier constructs a hyperplane between a category k and all of other categories. The other approach is “all-pairs” approach which builds $\frac{k(k-1)}{2}$ classifiers considering all pairs of categories. The final decision of the all-pairs approach is given by some voting method.

For SVM we used a word frequency in a document to employ as an attribute value for each word, since it has been shown to yield good performance for SVM based text categorization methods [4].

We applied SVM^{light} [8][†] in our experiments. Although many options are available in SVM^{light}, we simply used (-t, 0) option which denotes the linear SVM. We didn’t combine (-j, 1) option which introduces cost factors to be able to adjust the cost of false positive

[†]<http://svmlight.joachims.org/>

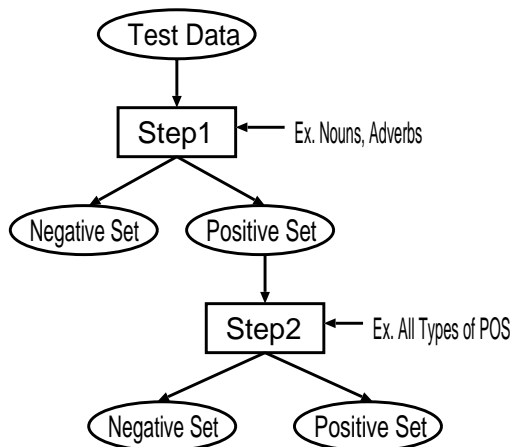


Fig. 2 Overview of our system when classifying test documents in the category “cocoa.”

vs. false negative [12], since the learning time and the categorization accuracy were not different from those of only (-t, 0) option.

3. Variable Cascaded Feature Selection

We propose a two step text categorization method with a variable cascaded feature selection (VCFS) using SVM to select suitable support vectors for each category in the training set.

Fig. 2 illustrates our idea. VCFS method consists of two steps. At step 1, SVM classifies test documents either in a positive or a negative set. At step 2, SVM again classifies the test documents which belong to the positive set of step 1. Therefore, we can expect high precision for the demand as described in Sect. 1.

Our system prepares 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000 words with the highest mutual information for each category on each POS combination as sets of features. Then, our system calculates F_1 measures for all possible pairs to select a pair of the best number of words and the best POS combination for each category at each step of the cascade using five-fold cross validation[†]. If some of pairs result in the same best F_1 measure, our system decides the best pair by the highest micro and macro averaged F_1 measures described in Sect. 4.2. F_1 measures may not be improved at step 2, even if we apply all possible pairs. In this case, our system selects the best pair for only step 1 and does not apply step 2. Thus, we call our method “variable” cascaded feature selection.

We treated {nouns, verbs}, {nouns, adjectives}, {nouns, adverbs}, {nouns, verbs, adjectives}, {nouns, verbs, adverbs}, {nouns, adjectives, adverbs}, and

[†]In five-fold cross validation, a training set is divided into five sets, one for a test set, and the other four for training sets. This process results in five individual F_1 measure and final features are selected by the average of the five F_1 measures.

{nouns, verbs, adjectives, adverbs} as POS combinations. We also added only nouns and all types of POS to the POS combinations. Henceforth, the notation {nouns, verbs, adjectives, adverbs} denotes that we extract words only from nouns, verbs, adjectives, and adverbs. Note that all POS combinations contain nouns, since nouns tend to be content words of a category compared with other POS.

Mutual information (MI) between a word w and a category c is defined as follows [3],[14].

$$MI(w, c) = \sum_{W \in \{w, \bar{w}\}} \sum_{C \in \{c, \bar{c}\}} P(W, C) \log \frac{P(W, C)}{P(W)P(C)} \quad (3)$$

MI becomes large when the occurrence of a word w is biased towards a category c . Consequently, the words with the highest mutual information in a category c can be expected to become keywords in the category c .

About the category “cocoa” of Reuters-21578 corpus, for example, our system selects {nouns, adverbs} at step 1 and also selects all types of POS at step 2. In this case, “seedpod,” “seedcase,” “pod,” “inflorescence,” “florescence,” “blossoming,” and so on which relate to “plants,” are selected as features when using {nouns, adverbs}, and “chocolate,” “cocoa,” and so on which relate to “drink,” are selected as features when using all types of POS.

4. Experiments

4.1 Data and Preprocessing

We evaluated VCFS method using Reuters-21578 corpus (ApteMod version) which is widely used in the text categorization research. Reuters-21578 corpus is divided into two sets, one for the training set and the other for the test set. All documents were extracted by eliminating unlabeled documents and selecting the categories which have at least one document both in the training and the test set. This process resulted in 90 categories. We obtained the training set of 7,769 documents for five-fold cross validation and the test set of 3,019 documents for evaluating VCFS method. The average number of categories assigned to a document was 1.3, and the largest number of categories assigned to a document was 15. Note that all documents were extracted without stemming and removing stop words.

Reuters-21578 corpus is known for a skewed category distribution [17]. The most common category has 2,877 training documents, but 82% of categories have less than 100 training documents, and 33% of categories have less than 10 training documents. Reuters-21578 corpus is also known for a rather direct correspondence between words and categories [6]. In the category “nickel,” for example, the occurrence of the word “nickel” in a document is a very good indicator.

Table 1 Relationships of an original word and the synonyms about the category “cocoa.”

Original Word	Synonyms
cocoa	chocolate
flowering	inflorescence, florescence, blossoming
pod	seedpod, seedcase

We tagged all documents with Brill’s POS tagger [1]^{††} and extracted words from {nouns, verbs}, {nouns, adjectives}, {nouns, adverbs}, {nouns, verbs, adjectives}, {nouns, verbs, adverbs}, {nouns, adjectives, adverbs}, {nouns, verbs, adjectives, adverbs}, only nouns, and all types of POS respectively.

We added synonyms of nouns in WordNet 1.7[†] to a set of features to represent documents before calculating mutual information, since both the micro and macro averaged F_1 measures described in Sect. 4.2 were improved, compared with the method without synonyms of nouns in WordNet 1.7. The micro-averaged F_1 measure was improved about 2.0% and the macro-averaged F_1 measure was improved about 10.0%. For example, Table 1 shows the relationships of an original word and the synonyms about the category “cocoa.” In Table 1, the column “Original Word” denotes a word appeared in the category “cocoa” of the training set and the column “Synonyms” denotes the synonyms of the word in WordNet 1.7.

4.2 Performance Measures

We used precision (Pr), recall (Re), and F_1 measure (F_1) in our experiments. These are calculated by the following formulas:

$$Pr = \frac{\text{number of categories found and correct}}{\text{number of total categories found}},$$

$$Re = \frac{\text{number of categories found and correct}}{\text{number of total categories correct}},$$

$$F_1 = \frac{2PrRe}{Pr + Re}.$$

We computed micro and macro averaged F_1 measures. The micro-averaged F_1 measure (miF_1) is obtained by first computing precision and recall for all categories and then using them to calculate F_1 measure. The macro-averaged F_1 measure (maF_1) is computed by first calculating F_1 measure for each category and then averaging them.

4.3 Results of VCFS Method

Table 2 and Table 3 show the comparisons of step 1 and step 2 on VCFS method. In Table 2 and Table 3, the column “Step” denotes each step of VCFS

Table 2 Comparison of Step 1 and Step 2. (micro-averaged)

Step	miPr (%)	miRe (%)	miF ₁ (%)
Step 1	90.5	79.2	84.4
Step 2	92.8	79.1	85.4

Table 3 Comparison of Step 1 and Step 2. (macro-averaged)

Step	maPr (%)	maRe (%)	maF ₁ (%)
Step 1	78.4	56.9	61.6
Step 2	84.2	56.0	64.8

method. Henceforth, the column “miPr,” “miRe,” and “miF₁” denote the micro-averaged precision, recall, and F_1 measure respectively and the column “maPr,” “maRe,” and “maF₁” denote the macro-averaged precision, recall, and F_1 measure respectively.

As shown in Table 2 and Table 3, we see that the macro-averaged F_1 measure is improved significantly at step 2, though the improvement of the micro-averaged F_1 measure is small.

Fukumoto et al. report that the micro-averaged F_1 measures of other SVM based text categorization methods were over 85% [5]. Therefore our micro-averaged F_1 measure is similar to them. However, our macro-averaged F_1 measure (64.8%) is significantly better than any reported F_1 measures (Fukumoto et al.(60.6%) [4], Yang et al.(52.5%) [17]). This indicates that VCFS method worked well to select suitable support vectors for each category in the training set.

Table 4 and Table 5 show what POS combination was used for each category at each step of the cascade. In Table 4, the column “POS Combination” denotes POS combination applied to VCFS method and the column “Step 1” and “Step 2” denote how many times the POS combination was applied at step 1 and step 2 respectively. In Table 5, the column “Step 1 → Step 2” denotes a pair of POS combinations applied to VCFS method and the column “Applied Frequency” denotes how many times the pair was applied. Henceforth, “a,” “n,” “v,” “adj,” and “adv” denote all types of POS, nouns, verbs, adjectives, and adverbs respectively.

As shown in Table 4, we see that only four POS combinations, which are {n, adv}, {n, v, adj}, {n, adj, adv}, and {n, v, adj, adv} are applied at step 1 and {n, v, adj} is most applied. About step 2, we see that all types of POS is most applied. Interestingly, our system doesn’t apply only nouns at both steps. From Table 5, we see that 19 categories are selected at step 2 and {n, adv} → a is most applied.

Table 6 shows the top 10 categories about the improvement of F_1 measure at step 2. Henceforth, the column “Category” denotes a category name and “Tr” denotes the number of training documents assigned to the category. From Table 6, we see that F_1 measures are improved not only on the categories assigned more than 100 training documents but also on the categories assigned less than 100 training documents.

^{††}<http://research.microsoft.com/users/brill/>

[†]<http://www.cogsci.princeton.edu/~wn/>

Table 4 POS combinations applied at each step.

POS Combination	Step 1	Step 2
a	0	5
n	0	0
{n, v}	0	3
{n, adj}	0	3
{n, adv}	14	1
{n, v, adj}	55	0
{n, v, adv}	0	2
{n, adj, adv}	8	3
{n, v, adj, adv}	13	2

Table 5 Pairs of POS combinations applied at each step.

Step 1 \rightarrow Step 2	Applied Frequency
{n, adv} \rightarrow a	3
{n, adv} \rightarrow {n, v}	1
{n, adv} \rightarrow {n, adj}	1
{n, v, adj} \rightarrow a	2
{n, v, adj} \rightarrow {n, v}	2
{n, v, adj} \rightarrow {n, adj}	1
{n, v, adj} \rightarrow {n, v, adv}	2
{n, v, adj} \rightarrow {n, adj, adv}	1
{n, v, adj} \rightarrow {n, v, adj, adv}	2
{n, adj, adv} \rightarrow {n, adj}	1
{n, adj, adv} \rightarrow {n, adv}	1
{n, v, adj, adv} \rightarrow {n, adj, adv}	2

Table 6 Top 10 categories on which F_1 measure of Step 2 was higher than that of Step 1. (Step 2/Step 1)

Category (Tr)	Pr (%)	Re (%)	F_1 (%)
oilseed (124)	100/5.3	100/100	100/10.0
cocoa (55)	100/27.3	83.3/100	90.9/42.9
money-supply (140)	60.0/17.2	60.0/100	60.0/29.4
reserves (55)	100/66.7	50.0/50.0	66.7/57.1
housing (16)	75.0/60.0	75.0/75.0	75.0/66.7
ipi (41)	75.0/64.3	75.0/75.0	75.0/69.2
orange (16)	100/87.5	63.6/63.6	77.8/73.7
barley (37)	100/88.9	66.7/66.7	80.0/76.2
rice (35)	94.4/85.0	73.9/73.9	82.9/79.1
yen (45)	100/50.0	16.7/16.7	28.6/25.0

For example, the category “cocoa” represents the characteristics of VCFS method clearly. About the category “cocoa,” both precision and F_1 measure are improved significantly at step 2, since the words “chocolate” and “cocoa” of step 2 worked well for choosing the exact test documents from the positive set of step 1. The words “chocolate” and “cocoa” were not selected as features at step 1.

About all categories, F_1 measures of 19 categories were improved at step 2 (14 categories assigned more than 100 training documents and 5 categories assigned less than 100 training documents) and F_1 measures of other categories remained the same.

We compared VCFS method with VCFS (k) method described in the following lines. Although VCFS method selects the best number of words with the highest mutual information for each category, VCFS (k) method simply uses k words with the highest mutual

Table 7 Comparison of VCFS method and VCFS (400) method. (micro-averaged)

Method	miPr (%)	miRe (%)	miF ₁ (%)
VCFS method	92.8	79.1	85.4
VCFS (400) method	92.5	75.8	83.3

Table 8 Comparison of VCFS method and VCFS (400) method. (macro-averaged)

Method	maPr (%)	maRe (%)	maF ₁ (%)
VCFS method	84.2	56.0	64.8
VCFS (400) method	77.6	46.6	56.0

Table 9 Top 10 categories on which F_1 measure of VCFS method was higher than that of VCFS (400) method. (VCFS method/VCFS (400) method)

Category (Tr)	Pr (%)	Re (%)	F_1 (%)
cocoa (55)	100/100	83.3/12.5	90.9/22.2
oilseed (124)	100/83.3	100/45.5	100/58.8
groundnut (5)	100/0.0	25.0/0.0	40.0/0.0
jobs (46)	92.3/60.0	57.1/25.0	70.6/35.3
reserves (55)	100/100	50.0/20.0	66.7/33.3
tea (9)	100/100	66.7/33.3	80.0/50.0
yen (45)	100/0.0	16.7/0.0	28.6/0.0
money-supply (140)	60.0/75.0	60.0/21.4	60.0/33.3
silver (21)	100/100	50.0/25.0	66.7/40.0
wpi (19)	100/100	40.0/20.0	57.1/33.3

information for each category. Table 7 and Table 8 show the comparison of VCFS method and VCFS (k) method. For VCFS (k) method we used $k = 400$, since the F_1 measure was the best.

From Table 7 and Table 8, we see that the macro-averaged F_1 measure of VCFS method is improved significantly compared with that of VCFS (400) method, though the improvement of the micro-averaged F_1 measure of VCFS method is small.

Table 9 shows how VCFS method worked on each category. Compared with VCFS (400) method, F_1 measures of 44 categories were improved (16 categories assigned more than 100 training documents and 28 categories assigned less than 100 training documents) and F_1 measures of other categories remained the same. This indicates that VCFS method was not affected by a category size because of selecting the best number of words with the highest mutual information for each category, though VCFS (400) method was affected by a category size, since it simply uses 400 words for each category.

4.4 Comparison of VCFS Method and Non-VCFS Methods

We compared VCFS method with Non-VCFS methods. For Non-VCFS methods we used ALL (Best) method and NOUN (Best) method. ALL (Best) method uses all types of POS and selects the best number of words with the highest mutual information for each category,

Table 10 Comparison of VCFS method and Non-VCFS methods. (micro-averaged)

Method	miPr (%)	miRe (%)	miF ₁ (%)
VCFS method	92.8	79.1	85.4
All (Best) method	92.4	71.7	80.8
Noun (Best) method	92.0	77.5	84.1

Table 11 Comparison of VCFS method and Non-VCFS methods. (macro-averaged)

Method	maPr (%)	maRe (%)	maF ₁ (%)
VCFS method	84.2	56.0	64.8
All (Best) method	70.1	36.2	45.2
Noun (Best) method	80.7	52.2	61.0

while NOUN (Best) method uses only nouns with the best number of words for which mutual information is the highest for each category. Although VCFS method selects the best POS combination for each category, Non-VCFS methods simply use all types of POS or only nouns.

Table 10 and Table 11 show the comparison of VCFS method and Non-VCFS methods. From Table 10 and Table 11, we see that the micro and macro averaged F_1 measures of VCFS method are improved significantly compared with those of ALL (Best) method. We also see that the macro-averaged F_1 measure of VCFS method is improved significantly compared with that of NOUN (Best) method, though the improvement of the micro-averaged F_1 measure of VCFS method is small. This indicates that VCFS method worked well to select suitable support vectors for each category in the training set.

We investigated how VCFS method worked on each category. Table 12 and Table 13 show the top 10 categories on which F_1 measure of VCFS method was higher than that of ALL (Best) method and NOUN (Best) method respectively. Also, Table 14 and Table 15 show the categories on which F_1 measure of VCFS method was lower than that of ALL (Best) method and NOUN (Best) method respectively.

Compared with ALL (Best) method, F_1 measure of VCFS method was higher on 55 categories (14 categories assigned more than 100 training documents and 41 categories assigned less than 100 training documents) and lower on a category assigned more than 100 training documents. F_1 measures of other categories remained the same.

Also, compared with NOUN (Best) method, F_1 measure of VCFS method was higher on 26 categories (9 categories assigned more than 100 training documents and 17 categories assigned less than 100 training documents) and lower on 8 categories (4 categories assigned more than 100 training documents and 4 categories assigned less than 100 training documents). F_1 measures of other categories remained the same.

Table 12 Top 10 categories on which F_1 measure of VCFS method was higher than that of ALL (Best) method. (VCFS method/ALL (Best) method)

Category (Tr)	Pr (%)	Re (%)	F ₁ (%)
nickel (8)	100/0.0	100/0.0	100/0.0
rapeseed (18)	100/0.0	75.0/0.0	85.7/0.0
lei (12)	100/0.0	66.7/0.0	80.0/0.0
tea (9)	100/0.0	66.7/0.0	80.0/0.0
soy-meal (13)	100/100	66.7/8.3	80.0/15.4
rice (35)	94.4/100	73.9/13.0	82.9/23.1
wpi (19)	100/0.0	40.0/0.0	57.1/0.0
ipi (41)	75.0/66.7	75.0/16.7	75.0/26.7
orange (16)	100/100	63.6/18.2	77.8/30.8
meal-feed (30)	100/100	63.6/18.8	77.8/31.6

Table 13 Top 10 categories on which F_1 measure of VCFS method was higher than that of NOUN (Best) method. (VCFS method/Noun (Best) method)

Category (Tr)	Pr (%)	Re (%)	F ₁ (%)
oilseed (124)	100/100	100/36.4	100/53.3
ipi (41)	75.0/50.0	75.0/25.0	75.0/33.3
cocoa (55)	100/87.5	83.3/41.2	90.9/56.0
income (9)	50.0/0.0	14.3/0.0	22.2/0.0
pet-chem (20)	100/0.0	10.0/0.0	18.2/0.0
meal-feed (30)	100/100	63.6/43.8	77.8/60.9
reserves (55)	100/100	50.0/33.3	66.7/50.0
yen (45)	100/100	16.7/9.1	28.6/16.7
wpi (19)	100/100	40.0/30.0	57.1/46.2
lead (15)	100/100	71.4/57.1	83.3/72.7

Table 14 A category on which F_1 measure of VCFS method was lower than that of ALL (Best) method. (VCFS method/ALL (Best) method)

Category (Tr)	Pr (%)	Re (%)	F ₁ (%)
dlr (131)	72.0/71.4	40.9/45.5	52.2/55.6

Table 15 Categories on which F_1 measure of VCFS method was lower than that of NOUN (Best) method. (VCFS method/NOUN (Best) method)

Category (Tr)	Pr (%)	Re (%)	F ₁ (%)
jobs (46)	92.3/92.9	57.1/61.9	70.6/74.3
sorghum (24)	83.3/75.0	62.5/75.0	71.4/75.0
gnp (101)	93.5/93.9	82.9/88.6	87.9/91.2
soybean (78)	67.9/78.3	61.3/58.1	64.4/66.7
dlr (131)	72.0/61.8	40.9/47.7	52.2/53.8
gold (94)	90.5/87.0	63.3/66.7	74.5/75.5
crude (389)	88.7/86.7	67.7/69.9	76.8/77.4
trade (369)	78.4/83.0	75.0/71.6	76.7/76.9

5. Conclusions

We have focused on the overfitting problem that SVM selects unsuitable support vectors for each category in the training set when focusing on all types of POS as input words and treating large numbers of training documents. To avoid the overfitting problem, we proposed the two step text categorization method with a variable cascaded feature selection (VCFS). For VCFS method we made use of the difference of words with the high-

est mutual information for each category on each POS combination.

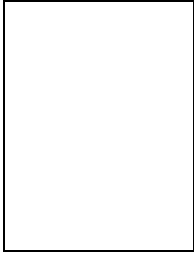
Through the experiments, we confirmed the validation of VCFS method compared with other SVM based text categorization methods, since our results showed that the macro-averaged F_1 measure (64.8%) of VCFS method was significantly better than any reported F_1 measures (Fukumoto et al. (60.6%) [4], Yang et al. (52.5%) [17]), though the micro-averaged F_1 measure (85.4%) of VCFS method was similar to them.

We compared VCFS method with Non-VCFS methods and confirmed that the micro and macro averaged F_1 measures of VCFS method were improved significantly compared with those of ALL (Best) method. We also confirmed that the macro-averaged F_1 measure of VCFS method was improved significantly compared with that of NOUN (Best) method, though the improvement of the micro-averaged F_1 measure of VCFS method was small.

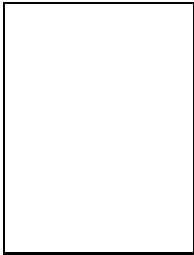
Although we used mutual information as a measure to select words to be used for text categorization in our experiments, we also confirmed the validation of VCFS method using other feature selection strategies such as χ^2 statistic.

References

- [1] E. Brill, "A simple rule-based part of speech tagger," Proc. 3rd Conference on Applied Natural Language Processing (ANLP'92), pp.152–155, 1992.
- [2] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," Proc. 7th International ACM Conference on Information and Knowledge Management, pp.148–155, 1998.
- [3] S. Dumais and H. Chen, "Hierarchical classification of web content," Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00), pp.256–263, 2000.
- [4] F. Fukumoto and Y. Suzuki, "Learning lexical representation for text categorization," Proc. NAACL'01 Workshop on Wordnet and Other Lexical Resources: Applications, Extensions and Customizations, pp.156–161, 2001.
- [5] F. Fukumoto and Y. Suzuki, "Manipulating large corpora for text categorization," Proc. 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP'02), pp.196–203, 2002.
- [6] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," Proc. 14th International Conference on Machine Learning (ICML'97), pp.143–151, 1997.
- [7] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," Proc. 10th European Conference on Machine Learning (ECML'98), pp.137–142, 1998.
- [8] T. Joachims, "Making large-scale support vector machine learning practical," Advances in Kernel Methods – Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola (eds.), MIT Press, Cambridge, MA, 1999.
- [9] T. Kudoh and Y. Matsumoto, "Chunking with support vector machines," Proc. 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'01), pp.192–199, 2001.
- [10] D. D. Lewis, Reuters-21578 text categorization test collection, Distribution 1.0, 1997.
- [11] Mainichi, CD Mainichi Shinbun 94. Nichigai Associates Co., 1995.
- [12] K. Morik, P. Brockhausen, and T. Joachims. "Combining statistical learning with a knowledge-based approach - A case study in intensive care monitoring," Proc. 16th International Conference on Machine Learning (ICML'99), pp.268–277, 1999.
- [13] F. Sebastiani, "Machine learning in automated text categorization," ACM Computing Surveys, vol.34, no.1, pp.1–47, 2002.
- [14] H. Taira and M. Haruno, "Feature selection in SVM text categorization," Proc. 16th National Conference on Artificial Intelligence, pp.480–486, 1999.
- [15] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, NY, 1995.
- [16] Y. Yang, "A study on thresholding strategies for text categorization," Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), pp.137–145, 2001.
- [17] Y. Yang and X. Liu, "A re-examination of text categorization methods," Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pp.42–49, 1999.



Takeshi Masuyama is a Ph.D. student, of Graduate School of Arts and Sciences, The University of Tokyo, Japan. His main research interests are natural language processing and information retrieval (especially text categorization). He received M.E. (2002) and B.E. (2000) degrees in the University of Yamanashi.



Hiroshi Nakagawa is a professor, of Information Technology Center, The University of Tokyo, Japan. His research interests are computational linguistics, natural language processing, information retrieval, text mining, language interface, and so on. He received Ph.D. (1980), M.E. (1977), and B.E. (1975) degrees in the University of Tokyo. During 1990–1991, he was a visiting researcher at CSLI, Stanford University in U.S.A.