ISSN 0918-2802

Technical Report

Text categorization based on weighted inverse document frequency

Tokunaga, Takenobu
Iwayama, Makoto[1]

94-TR0001 March 1994

Department of Computer Science
Tokyo Institute of Technology
Ôokayama 2-12-1 Meguro Tokyo 152, Japan

[1]Advanced Research Laboratory, Hitachi Ltd.

Abstract

This paper proposes a new term weighting method called *weighted inverse document frequency* (WIDF). As its name indicates, WIDF is an extension of IDF (inverse document frequency) to incorporate the term frequency over the collection of texts. WIDF of a term in a text is given by dividing the frequency of the term in the text by the sum of the frequency of the term over the collection of texts. WIDF is applied to the text categorization task and proved to be superior to the other methods. The improvement of accuracy on IDF is 7.4% at the maximum.

1 Introduction

Text categorization is the classification of texts with respect to a set of predefined categories. The categorization task has typically been done by human experts. However, as the number of texts increases, it becomes difficult for humans to consistently categorize them. Therefore, automatic text categorization is an essential technology for intelligent information systems, and has received much attention in recent years [1, 2, 3, 4, 5, 6].

There are several approaches to text categorization, such as decision rule based [5], knowledge base based, text similarity based and so on. In this paper, we focus on text categorization based on text similarity. The text categorization task based on text similarity requires the following two subtasks:

- **Similarity measurement:**
  to calculate the similarity between an incoming text and the pre-categorized texts,
- **Category assignment:**
  to assign the category that is most similar to the incoming text.

For each subtask, there are several options we can take. Figure 1 shows the structure of text categorization and their options. Curly brackets (" { ") denote that one of the components is necessary, and square brackets (" [")denote that all the components are necessary.

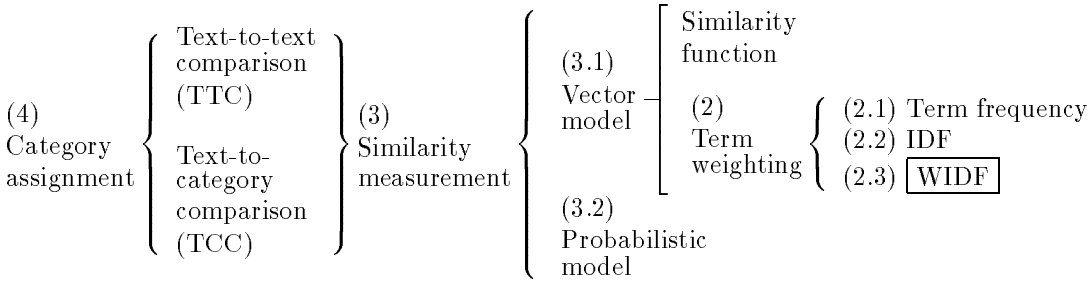

Fig. 1 Structure of text categorization

We propose a new term weighting method which is used in the vector model. The proposed method is called *weighted inverse document frequency* (WIDF). WIDF is an extension of inverse document frequency (IDF) to incorporate the term frequency over the collection of texts. We have previously proposed a similar measure to calculate the degree of feature salience in concepts [7]. This measure of salience proved useful in the area of natural language processing [7, 8, 9]. In this paper, we show that the idea from our previous work is also applicable to the text categorization task and prove its superiority to other measures.

The text categorization task is more suitable for evaluating measures than the information retrieval task. In case of text categorization, a set of categories is predefined and we would able to assign the categories to texts in a collection. This collection of pre-categorized texts enable us to evaluate measures fully automatic way. On the other hand, in case of information retrieval, we have problems in defining *relevance* of documents which would differ from user to user. Therefore it is more difficult to obtain a collection for evaluation.

For the evaluation with text categorization, we used several collections of texts which were excerpted from *Gendaiyôgo no Kisotisiki* (Dictionary of Japanese Contemporary Terms). The size of the collections range from 185 texts to 6246 texts. According to figure 1, we have eight possible text categorization

systems depending on the choices of the components. The experiments show that the system that adopts WIDF and TCC gives the best result among these systems.

Figure 1 also shows the structure of this paper. The numbers in the parentheses denote the corresponding sections of this paper. Section 2 briefly reviews term weighting methods and defines WIDF. Section 3 explains the similarity measurement using both the vector model and the probabilistic model. Section 4 describes two category assignment methods used in the experiments. The details of the experiments are described in section 5. Finally, section 6 summarizes the paper and discusses future research directions.

2 Term Weighting

Term weighting is one of the important issues in text categorization. Originally, term weighting has been widely investigated in information retrieval [10, 11]. This section briefly reviews the previous term weighting methods, and proposes a new method called *weighted inverse document frequency* (WIDF).

2.1 Term Frequency

Term frequency is the simplest measure to weight each term in a text. In this method, each term is assumed to have importance proportional to the number of times it occures in a text [12]. The weight of a term t in a text d is given by

$$W(d,t) = \mathrm{TF}(d,t), \tag{1}$$

where $\mathrm{TF}(d,t)$ is the term frequency of the term t in the text d. Term frequency is known to improve recall in information retrieval, but does not always improve precision. Because frequent terms tend to appear in many texts, such terms have little discriminative power. In order to remedy this problem, terms with high frequency are usually removed from the term set. Finding optimal thresholds is the main concern in this method.

2.2 Inverse Document Frequency

While term frequency concerns term occurrence within a text, inverse document frequency (IDF) concerns term occurrence across a collection of texts. The intuitive meaning of IDF is that terms which rarely occur over a collection of texts are valuable. The importance of each term is assumed to be inversely proportional to the number of texts that contain the term [13]. The IDF factor of a term t is given by

$$\mathrm{IDF}(t) = \log(N/df(t)), \tag{2}$$

where N is the total number of texts in the collection and $df(t)$ is the number of texts that contain the term t. This definition follows Salton's definition ([11]:p280). Since IDF represents term specificity, it is expected to improve the precision. Salton proposed to combine term frequency and IDF to weight terms, and showed that the product of them gave better performance [14]. The combination weight of a term t in a text d is given by

$$W(d,t) = \mathrm{TF}(d,t) \cdot \mathrm{IDF}(t). \tag{3}$$

The factors $\mathrm{TF}(d,t)$ and $\mathrm{IDF}(t)$ would contribute to improve the recall and the precision respectively.

2.3 Weighted Inverse Document Frequency

A drawback of IDF is that all the texts that contain a certain term are treated equally. That is, IDF does not distinguish between one occurrence of a term in a text and many. Let us consider the example in table 1. d (column) stands for a text and t (row) stands for a term. The intersection of d_i and t_j stands for the term frequency of the term t_j in the text d_i.

Table 1 Example of Collection

	d_1	d_2	d_3	d_4	d_5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t_x	2	50	3	2	4
t_y	3	2	3	2	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Following Eq. (2), both the IDF(t_x) and IDF(t_y) are equal to 0, since they appear in all the texts d_1 through d_5. However, the frequency distributions of t_x and t_y over d_1 through d_5 are quite different. IDF is not able to reflect this difference because $df(t)$ in Eq. (2) concerns only if the term t is contained in the text or not (i.e. binary counting); it does not take into account the frequency of the term t. We extend IDF to incorporate the term frequency over the collection of texts.

IDF assumes that the importance of a term is inversely proportional to the number of texts that contain the term, therefore, the essential part of Eq. (2) is the factor $1/df(t)$ [1]. Our proposal weights this factor with the term frequency. For example, $1/df(t)$ of t_x in table 1 becomes

$$\frac{1}{1+1+1+1+1}$$

for all the texts. We weight each of these "1" by the frequency of each term, which becomes

$$\frac{50}{2+50+3+2+4}$$

for the case of d_2. The factor is called *weighted inverse document frequency* (WIDF). Note that unlike IDF, WIDF differs for each text, $d_1 \ldots d_5$. Formally, WIDF of a term t in a text d is given by

$$\mathrm{WIDF}(d,t) = \frac{\mathrm{TF}(d,t)}{\sum_{i \in D} \mathrm{TF}(i,t)} \tag{4}$$

where TF(d,t) is the term frequency of the term t in the text d and i ranges over the texts in the collection D. WIDF of a term t sums up to one over the collection of texts. In other words, WIDF corresponds to the normalized term frequency over the collection. WIDF is a measure similar to the authors' previous work [7], in which a WIDF-like measure is used for calculating the salience of features in concepts.

Using WIDF, the weight of a term t in a text d is given by

$$W(d,t) = \mathrm{WIDF}(d,t). \tag{5}$$

Because Eq. (4) already includes TF(d,t) in the numerator, we do not have to multiply the factor TF(d,t) for the recall as in Eq. (3). WIDF itself includes factors that would contribute to both recall and precision. The comparison of the performance of Eq. (3) and Eq. (5) is given in section 5.

3 Similarity Measurement

3.1 Vector Model

Using a term weighting method, texts would be represented by term vectors of the form

$$V_d = (w_1, w_2, \ldots, w_n) \tag{6}$$

where the element w_i corresponds to the weight of the term i. Given the vector representations of texts as in Eq. (6), a similarity between two texts would be obtained by element-wise comparison of the vectors.

[1] We conducted preliminary experiments of text categorization to see the influence of replacing Eq. (2) with $1/df(t)$. We could not find significant difference.

There are many ways to measure the similarity between two vectors [10]. In this paper, we adopt the *Jaccard* function, which is the normalized inner product of two vectors as shown in Eq. (7).

$$Sim(V_i, V_j) = \frac{\sum_{k=1}^{n}(w_{ik} \cdot w_{jk})}{\sum_{k=1}^{n} w_{ik} + \sum_{k=1}^{n} w_{jk} - \sum_{k=1}^{n} w_{ik} \cdot w_{jk}}, \tag{7}$$

where w_{ik} means the k-th element of the term vector V_i. The greater the value of $Sim(V_i, V_j)$ is, the more similar these two texts are. Thus, given a set of pre-categorized texts and a new text, we could categorize the text into a category which is assigned to the most similar pre-categorized text.

3.2 Probabilistic Model (Bayesian Model)

From the viewpoint of probability theory, we can calculate the probability that a text d is categorized into a category C_i, that is $P(C_i|d)$.

$$P(C_i|d) = \sum_t P(C_i|t,d)P(t|d), \tag{8}$$

where t is a term that ranges over the vector elements of C_i and d. If we assume conditional independency between C_i and d given t, that is $P(C_i|t,d) = P(C_i|t)$, we obtain Eq. (9).

$$P(C_i|d) = \sum_t P(C_i|t)P(t|d). \tag{9}$$

Using Bayes' rule, we finally obtain Eq. (10).

$$P(C_i|d) = P(C_i)\sum_t \frac{P(t|C_i)P(t|d)}{P(t)}. \tag{10}$$

This formulation is different from the one proposed in [3, 15]. The details of this formulation is discussed elsewhere [16]. Here $P(t|C_i)$ is the probability that a randomly selected term in the category C_i is the term t. $P(t|d)$ is the probability that a randomly selected term in the text t is the term t. $P(t)$ and $P(C_i)$ are the prior probabilities of terms and categories respectively. All these probabilities are estimated from the training data set. A text d will be categorized into the category C_i which has the highest probability of $P(C_i|d)$. This probabilistic model will be compared with the vector model in section 5.

4 Category Assignment

Given a set of pre-categorized texts and a new text, there are several approaches to assigning a category to the text. We consider the follwing two approaches in this paper. The first approach is based on text-to-text comparison, also called k-nearest neighbor (k-NN) or memory based reasoning (MBR) [4, 17]. The second one is based on text-to-category comparison.

1. **Text-to-Text Comparison (TTC):**
   In the k-nearest neighbor method, all the similarity between a new text and each of pre-categorized texts are calculated, then k most similar texts vote on the category.

2. **Text-to-Category Comparison (TCC):**
   In the second method, all the texts that belong to the same category are bundled into a single text and the similarities between a new text and each of these bundled texts are calculated.

Note that while TTC requires N comparisons, TCC requires only C comparisons, where N and C stand for the number of pre-categorized texts and the number of categories respectively (usually $N \gg C$). This means that TCC is more computationally efficient than TTC. In addition, TCC is expected to have better generalization ability of categorization by bundling up the texts of the same category. That is, TCC would have higher accuracy for categorizing new texts. On the other hand, TTC especially with smaller k tends to suffer the effect of noise that exists in the training data. This phenomenon is known as over-fitting to the training data [17]. The performance of both the methods is evaluated in the next section.

5 Experiments

This section describes experiments to evaluate the performance of WIDF. As shown in figure 1, we have several options to construct a text categorization system. The experiments are conducted for the eight possible combinations as shown in table 2.

Table 2 Cases of Experiments

(E1)	TTC+Vector model (TF)	(E5)	TCC+Vector model (TF)
(E2)	TTC+Vector model (TF $\cdot$ IDF)	(E6)	TCC+Vector model (TF $\cdot$ IDF)
(E3)	TTC+Vector model (WIDF)	(E7)	TCC+Vector model (WIDF)
(E4)	TTC+Bayesian model	(E8)	TCC+Bayesian model

In all the cases that include vector model, Jaccard function (Eq. (7)) is used for the similarity function.

5.1 Data and Preprocessing

As data for evaluation, we use *Gendaiyôgo no Kisotisiki* (Dictionary of Japanese Contemporary Terms) of 1992 version (GK for short hereafter) [18]. GK contains 18,476 entries of contemporary terms, which are categorized into 149 minor categories. These minor categories are further categorized into 13 major categories as shown bellow. The number in the parentheses stands for the number of the minor categories that belongs to each major category.

0. Special issues of 1992 (8)
1. International affairs (20)
2. Politics, Diplomacy, Law (9)
3. Economics, Industry, Management, Labour (22)
4. Information, Communication (6)
5. Science, Technology (16)
6. Society, Living (20)
7. Medical science (8)
8. Culture, Art (16)
9. Fashion (11)
10. Sports, Leisure (11)
11. Imported terms in 1992 (1)
12. Acronyms in 1992 (1)

The major category (0) is excluded because the entries in the category (0) are specific to 1992 and contains diverse topics. The entries in major categories (11) and (12) are also excluded because they all have very short contents, namely less than 20 characters on average. These exclusions reduce the number of entries to 10,135. The length of the remaining texts varies from 13 to 1938 characters, on average 287 characters. We use the minor categories for categorization, that is, we have 139 predefined categories. Each text is assigned to only one of these categories.

Since GK is written in Japanese, morphological analysis is necessary in order to extract terms from the texts. Japanese morphological analyzer JUMAN [19] which was developed at Kyoto University and Nara Institute of Science and Technology is used to segment and tag the texts. We evaluated the performace of JUMAN with about 1,000 sentences of newspaper articles. According to the evaluation, the error rate of JUMAN is about 10% in segmentation and 30% in part of speech tagging. Segmentation means identifying the boundaries between words, since no spaces are placed between words in Japanese. From the analyzed texts, nouns and unknown words are extracted as terms. Since most unknown words are nouns, in particular proper nouns that could characterize the text very well, unknown words are added to the term set of the texts. This procedure creates the vectors of term frequency for each text. The number of terms varies from 3 to 546. In order to resolve the variation of the number of terms in a text, the elements of the vectors (term frequencies) are normalized so that all the elements sum up to one. By using these normalized term frequencies, we calculate the measures, namyly, IDF and WIDF.

In order to see how the number of terms in a text and the number of texts in a category affect categorization, the collections of the texts are selected by using the following two thresholds:

- the number of terms in a text (60, 80, 100, 120),
- the number of texts that belong to the same category (20, 30, 40).

The combination of the above thresholds gives 12 (4×3) collections of texts. Table 3 shows the number of texts and the number of categories of the collections. Each collection is referred to as "collection XX-YY" hereafter, where XX stands for the first threshold and YY does the second one. For instance, the collection 100-20 stands for the collection in which each text has at least 100 terms, and each category includes at least 20 texts. According to table 3, the collection 100-20 includes 2240 texts, each of which is categorized into one of 66 categories.

Table 3 Number of texts and categories of each collection

		No. of texts in a category		
		20	30	40
No. of terms in a text	60	6246 / 120	5819 / 102	4943 / 77
	80	4048 / 95	3431 / 70	2773 / 51
	100	2240 / 66	1546 / 37	761 / 14
	120	1072 / 39	423 / 11	185 / 4

(No. of texts / No. of categories)

5.2 Text-to-Text Comparison (TTC)

Since the computational cost of TTC is expensive, TTC is applied only to the collection 100-20. The similarity between every pair of texts in the collection is calculated by using the vector model (Eq. (7)) with three term weighting methods (Eq. (1), (3) and (4)), and by using the Bayesian model (Eq. (10)). k is varied from 1 to 11 by 2. The evaluation is done by 4-fold cross validation, that is, a collection is divided into four subsets and one of them is used as a test set and the others as a training set.

The accuracy of categorization is calculated by

$$\text{Accuracy} = \frac{\text{the number of texts that are assigned to the correct category}}{\text{the number of total texts in the collection}} \tag{11}$$

The average accuracy over all 4 test sets is calculated. Table 4 shows the result, which corresponds to (E1) through (E4) in table 2.

Table 4 Accuracy of categorization with collection 100-20 (TTC) [%]

k	1	3	5	7	9	11
(E1) TF	45.8	49.5	55.2	57.7	59.2	59.4
(E2) TF $\cdot$ IDF	60.9	65.3	70.6	72.0	73.0	73.1
(E3) WIDF	34.8	39.1	46.8	52.5	56.5	59.0
(E4) Bayes	63.7	68.0	71.7	74.0	75.0	75.9

5.3 Text-to-Category Comparison (TCC)

In TCC, the texts of training sets that belong to the same category are bundled into a single text. Therefore, we have one text per category in the training data. Next the similarity between each text in the test set and each of training texts is calculated. A text in the test set is categorized into the category that the most similar training text belongs to. The accuracy is evaluated by 4-fold cross validation. In addition, the recall and the precision are also evaluated by 4-fold cross validation. Recall and precision are calculated by the following equations:

$$\text{Recall} = \frac{\text{the number of texts that are assigned to the correct category}}{\text{the number of total texts in the category}},$$

$$\text{Precision} = \frac{\text{the number of texts that are assigned to the correct category}}{\text{the number of total texts that are assigned to the category}}.$$

Recall and precision are calculated for each category and the average over the collection is obtained. Table 5 shows the result, which corresponds to (E5) through (E8) in table 2.

Table 5 Result of categorization (TCC) [%]

	Recall / Precision				Accuracy			
	(E5)	(E6)	(E7)	(E8)	(E5)	(E6)	(E7)	(E8)
Collection	TF	TF · IDF	WIDF	Bayes	TF	TF · IDF	WIDF	Bayes
60-20	70.4 / 74.4	76.3 / 74.5	79.6 / 78.4	72.0 / 79.1	67.7	75.0	79.0	76.1
60-30	72.6 / 76.4	77.4 / 76.7	80.9 / 80.2	75.2 / 79.9	70.9	76.5	80.4	77.8
60-40	74.4 / 79.8	80.6 / 80.6	84.4 / 84.4	78.7 / 82.6	73.4	80.2	84.3	80.8
80-20	69.1 / 74.9	74.6 / 73.8	79.4 / 78.7	71.3 / 77.8	67.7	73.8	79.2	75.2
80-30	72.1 / 78.8	79.1 / 78.7	84.0 / 83.8	77.8 / 82.2	71.1	78.2	83.6	79.7
80-40	76.3 / 80.6	81.2 / 80.9	86.7 / 86.7	80.5 / 84.0	76.6	80.9	86.7	82.2
100-20	70.4 / 77.7	77.0 / 76.2	81.4 / 80.3	71.8 / 79.0	70.0	76.2	81.3	75.3
100-30	78.4 / 81.9	84.8 / 84.0	89.6 / 89.6	81.9 / 86.8	78.2	84.3	89.6	83.6
100-40	89.0 / 90.2	92.7 / 93.1	95.6 / 96.3	93.1 / 94.5	89.2	92.6	95.5	93.2
120-20	75.0 / 80.2	81.5 / 79.5	86.9 / 86.9	79.0 / 83.5	73.9	79.1	86.5	80.5
120-30	91.2 / 91.5	92.5 / 92.3	96.0 / 96.3	91.8 / 93.6	90.5	91.7	95.5	91.7
120-40	90.9 / 91.8	88.1 / 88.4	93.7 / 93.8	91.8 / 91.9	90.8	88.1	93.5	91.9

As we expected, TCC gives better results than TTC in all the cases except the Bayesian model with $k = 11$. This superiority of TCC proves that TCC has better generalization ability than TTC. In addition, TCC is advantageous with respect to the computational cost.

In TCC, WIDF gives the best result. The superiority of WIDF to TF · IDF ranges from 2.9% to 7.4% in accuracy. The superiority to Bayesian ranges from 1.6% to 6.0%. These differences are significant on the basis of the standard error analysis [17][2].

WIDF gives better results also in recall and precision. In general, recall and precision are mutually exclusive factors, that is, we could have high recall value at the cost of precision and vice versa. The balance of recall and precision is an important factor for applications. From this point of view, WIDF is superior to the Bayesian model, because the difference of recall and precision is quite large in the Bayesian model compared to the vector model with WIDF.

We made several collections by varing the two threshold, that is, the number of categories in a collection and the number of terms in a text. A tendency observable in table 2 is that the smaller the number of categories is, the higher the accuracy is. Also the larger the number of the terms in a text is, the higher the accuracy is.

6 Concluding Remarks

We proposed a new term weighting method, called weighted inverse document frequency (WIDF). WIDF of a term in a text is calculated by dividing the frequency of the term in the text by the sum of all the frequency of the term over the collection of texts. We applied this method to the text categorization task. We used collections of 200–6000 texts, each of which has 60–120 terms for the experiments, which showed that WIDF provides better accuracy than TF · IDF by 7.4% and Bayesian by 6.0% at the maximum. As future research directions, we plan to conduct experiments with other collections of texts. Also we will apply WIDF to other applications such as text clustering.

References

[1] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, Vol. 35, No. 12, pp. 29–38, 1992.

[2] M. J. Blosseville, G. Hébrail, M. G. Monteil, and N. Pénot. Automatic document classification: Natural language processing, statistical analysis, and expret system techniques used togeter. In *Proceedins of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 51–58, 1992.

[2] The standard error of the error rate is estimated by $\mathrm{SE} = \sqrt{\frac{E(1-E)}{n}}$ where E is the error rate on n test cases.

[3] D. D. Lewis. An evaluation of phrasal and clustered representations of a text categorization task. In *Proceedins of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 37–50, 1992.

[4] B. Masand, G. Finoff, and D. Waltz. Classifying news stories using memory based reasoning. In *Proceedins of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 59–65, 1992.

[5] C. Apte, F. Damerau, and S. M. Weiss. Automatic learning of decision rules for text categorization. Research Report RC19979(82518), IBM, 1993.

[6] D. Biber. Using register-diversified corpora for general language studies. *Computational Linguistics*, Vol. 19, No. 2, pp. 219–241, 1993.

[7] M. Iwayama, T. Tokunaga, and H. Tanaka. A method of calculating the measure of salience in understanding metaphors. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 298–303, 1990.

[8] T. Pattabhiraman and N. Cercone. Communicating properties using salience-induced comparisons. In *14th Annual Conference of the Cognitive Science Society*, pp. 1032–1037, 1992.

[9] L. F. Rau. Calculating salience of knowledge. In *14th Annual Conference of the Cognitive Science Society*, pp. 564–569, 1992.

[10] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

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

[12] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, Vol. 1, No. 4, pp. 307–319, 1957.

[13] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, Vol. 28, No. 1, pp. 11–21, 1972.

[14] G. Salton and C. S. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, Vol. 29, No. 4, pp. 351–372, 1973.

[15] N. Fuhr. Models for retrieval with probabilistic indexing. *Information Processing & Management*, Vol. 25, No. 1, pp. 55–72, 1989.

[16] M. Iwayama and T. Tokunaga. A probabilistic model for text categorization: Based on single random variable with multiple values. unpablished manuscript, 1994.

[17] S. M. Weiss and C. A. Kulikowski. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems.* Morgan Kaufmann, 1991.

[18] *Gendaiyôgo no Kisotisiki* (Dictionary of Contemporary Terms). Jiyûkokuminsya, 1992. (in Japanese).

[19] Y. Matsumoto, et al. *JUMAN Users Manual.* Kyoto University and Nara Institute of Science and Technology, 1993.