Web Page Classification Based on k-Nearest Neighbor Approach

Oh-Woog Kwon and Jong-Hyeok Lee

Dept. of Computer Science and Engineering
Pohang University of Science and Technology
San 31 Hyoja Dong, Pohang, 790-784, Korea
Email: ohwoog@kle.postech.ac.kr, jhlee@postech.ac.kr

Abstract

Automatic categorization is the only viable method to deal with the scaling problem of the World Wide Web. In this paper, we propose a Web page classifier based on an adaptation of k-Nearest Neighbor (k-NN) approach. To improve the performance of k-NN approach, we supplement k-NN approach with a feature selection method and a term-weighting scheme using markup tags, and reform document-document similarity measure used in vector space model. In our experiments on a Korean commercial Web directory, our proposed methods in k-NN approach for Web page classification improved the performance of classification.

Keywords: text categorization, Web page classification, k-nearest neighbor approach, feature selection, term weighting scheme, similarity measure.

1 Introduction

Text categorization is the automated assigning of predefined subject categories to documents. As the amount of online texts such as Web pages dramatically increases, the demand for text categorization to aid efficient retrieval, by filtering out unsound Web pages and management of the World Wide Web (WWW) is increasing. Traditionally, this task is performed manually by domain experts. However, human categorization is unlikely to keep pace with the rate of growth of the WWW. Hence, as the WWW continues to increase, the importance of automatic Web page categorization becomes obvious. Moreover, automatic categorization is much cheaper and faster than human categorization.

Many different learning-based approaches have been applied to the text categorization task, including k-Nearest Neighbour (k-NN) approach [2, 10, 11, 12, 16, 17, 18, 19], Bayesian probabilistic approaches [1, 5, 7, 9, 10, 11], inductive rule learning [3], Support Vector Machines [13, 14, 15], neural networks [18], and decision trees [7]. The recent literature of Yang and Liu helps us to construct a model, which reported that the k-NN classifier performed comparably or better, compared with other well-known approaches [18]. In other literatures [10, 17], k-NN is known as one of the top-performing approaches in text categorization task.

This paper also addresses three issues concerning k-NN classification applied into Web page classification: feature selection, term weighting, and document-document similarity. First, earlier k-NN classifiers were given a training document, with every term appearing in the document after the removal of stop-words, but our k-NN classifier's training document contains only terms related to the categories assigned to the document through feature selection. HyperText Markup Language (HMTL) tags of a Web page is closely related to the content of the Web page, so that the Web page must be represented with information composed of terms and HTML tags annotated with terms. Secondly, we propose a term weighing scheme using HTML tags annotated with terms, as well as traditional term statistics. Because k-NN classifier determines the promising categories for a new document using k-nearest samples, document-document similarity measure might have a critical effect on the performance of a k-NN classifier. Third, to select k-nearest neighbours as more relevant training documents, we propose a similarity measure modified from a traditional similarity used in a vector space model.

In this paper, we have conducted experiments on a Web site test collection. The Web sites and hierarchical categories of the test collection are serviced in *HanMir* (http://www.hanmir.com), which is a popular Web search engine in Korea.

Proceedings of the 5th International Workshop Information Retrieval with Asian Languages

The rest of this paper is organized as follows. Section 2 gives a description of general k-NN approach. Section 3 explains use of feature selection method in k-NN approach. In Section 4 we describe proposed term weighting scheme using HTML tags. Section 5 gives a detailed explanation for modified similarity measure based on vector space model. Section 6 gives a description of our test collection. We present and evaluate the empirical results achieved by our proposed k-NN approach in Section 7. Finally, concluding remarks are given in Section 8.

2 k-NN Approach

The k-NN approach can be broken down in two steps. Given a test document, the first step is to find k nearest samples among the training documents, using document-document similarity. The similarity score of each neighbor document to the test document is used as the weight of the categories pre-assigned to the training document. The second step is to estimate the likelihood of each category by summing the weight of the category of the k nearest documents as follows [10, 12, 17, 18, 19]:

$$P(C_j \mid D_x) \approx \sum_{D_i \in \{k-nearest\ documents\}} sim(D_x, D_i) P(C_j \mid D_x) \qquad (1)$$

where $sim(D_x, D_i)$ is the similarity between the test document D_x and a training document D_i, and $P(C_j \mid D_x) \in \{0,1\}$ is the classification for the document D_i with respect to category C_j ($P(C_j \mid D_x) = 1$ for YES, and $P(C_j \mid D_x) = 0$ for NO).

In this paper to improve the performance of k-NN approach, we supplement the k-NN approach with two factors (feature selection and use of HTML tags) and also modify the similarity measure. In our previous research [12], we proposed additional factors and evaluated their effectiveness on two test collections as follows: the Reuter-21578 collection, widely used in text categorization, and a Web site collection which is the prototype version for Web site classification of *HanMir* [12]. In here, we briefly explain feature selection and the use of HTML tags and also show the experimental results from our previous research. Next, we explain a modified similarity measure newly proposed in this paper. In this paper, performance of k-NN classifier is measured by micro-averaging recall and precision, which are widely used to evaluate overall performance across an entire set of categories in text categorization researches [5, 6]. To compact recall and precision to a single measure, we use the micro-averaging breakeven point, which is the first point where recall equals precision.

3 Feature Selection

A common k-NN classifier represents each training document as the terms occurred in the document after the removal of stop-words [1, 9, 11, 12, 13], but we cannot tell that all the terms are related to the categories of the training document. The terms not related to the categories of the training document must be noise in the text categorization task. Hence, our k-NN classifier reduces the noise terms using a feature selection method unlike past k-NN classifiers. To reduce the noise of a training document, we first for each category, select index terms (features) predicting the occurrence of that category, and remove all terms not selected as features of the categories assigned to the training document. The ideal effect of our feature selection method is the removal of all noise terms that are unrelated to the categories assigned to the training document in which the index terms occur. Because feature selection reduces the dimension of the vectors, our classifier can find the k-nearest training document faster. Also, training documents having the same categories are represented with similar term vectors, so that they can be closely located in term vector space. This result of feature selection demonstrates a favorable effect on text categorization.

For feature selection, we choose expected mutual information (EMI) and mutual information (MI) measurements that are widely used in text categorization, pattern recognition and machine learning [4, 5]. We select an index term as a feature for a category if both values of EMI and MI between the term and the category are above user-defined thresholds of EMI and MI respectively. EMI is acquired by using both the similarity and the dissimilarity between an index term and a category, as defined in Formula (2) [4, 5].

$$EMI(T_i, C_k) = \sum_{a=0,1} \sum_{b=0,1} P(T_i = a, C_j = b) \log \frac{P(T_i = a, C_j = b)}{P(T_i = a) \times P(C_j = b)} \qquad (2)$$

The window of co-occurrence is fixed within a document in Formula (4) and (5). According to Formula (4), we can acquire not only positive features, but also negative features for each category. Because negative features are not useful in a k-NN classifier, we select only positive features using MI defined by Formula (3) [4].

$$MI(T_i, C_k) = \log \frac{P(T_i = 1, C_j = 1)}{P(T_i = 1) \times P(C_j = 1)} \qquad (3)$$

In EMI, if the similarity between a term and a category is much higher than the dissimilarities, the term is selected as a positive feature for the category. The dissimilarities between the term and the category indicate that the probability that the term can be selected as positive feature for other categories is very high. Consequently, the positive features extracted by EMI tend to predict fewer categories than the positive features extracted by MI. Generally, the fewer categories a term predicts, the better. Hence, we did not use only MI to acquire the positive features for each category.

The use of feature selection in k-NN approach improved micro-averaging breakeven point by 3.3% on Reuter 21578

and 7.5% on the Web site collection in our previous research [12].

4 Term Weighting Scheme Using HTML Tags

For readers' attention, authors of Web pages often annotate important words, phases, sentences, or paragraphs with special HTML tags such as title tag, headline tag, bold tag, blink tag, etc. Therefore, the expression power of a tag may actually be used to indicate the importance of terms annotated with the tag for content representation of Web pages. To define the expression power of tags, we first divide tags into several groups according to their estimated expression power, and then assign a user-defined weight to each group. If authors overuse some tags to compose Web pages compared with common case, weights of the abused tags must be decreased because the tags are not used for the purpose of emphasis. We can easily detect the abused tags by comparison between the proportion of the frequency of a tag to the total frequency of all tags in a Web page and the proportion in a collection. To decrease the weights of an abused tag, we divide the weight of the tag by overused degree (= percentage of the tag in the collection / percentage of the tag in a Web page). We can define the importance of a markup tag Tag_k in a document D_i by Formula (4) using the distribution ratio of the tag and the user-defined weight of the tag.

$$Weight(Tag_k) = UDW(Tag_k) \times DR(Tag_k, D_i) \quad (4)$$

where

$UDW(Tag_k)$ is the user-defined weight of markup tag Tag_k,

and $DR(Tag_k, D_i) = \begin{cases} 1, & \text{if } \beta \le \lambda, \\ \lambda / \beta, & \text{otherwise.} \end{cases}$

where $\beta = \dfrac{\text{the frequency of tag } Tag_k \text{ in document } D_i}{\text{the total frequency of all tags in document } D_i}$,

and $\lambda = \dfrac{\text{the frequency of tag } Tag_k \text{ in a collection}}{\text{the total frequency of all tags in a collection}}$.

In Formula (4), function $DR(Tag_k, D_i)$ is to decrease the user-defined weight of tag Tag_k when the tag Tag_k is overused in the document D_i.

So far, we define only weighting scheme for markup tags in an HTML document. To assign weight to terms, the weighting scheme of markup tag is integrated to a term weighting scheme based on term frequency and inverse document frequency. Now, weight WT_pD_i of term T_p in document D_i is defined in Formula (5). In Formula (5), when a term is annotated with several nested markup tags, we select the tag with maximum weight among them. Simply explaining for Formula (5), if a term occurs in a tag with weight a, we count the term frequency a times, instead of 1, and then the term weight is calculated by a logarithm formula of ***tf · idf***.

$$W_{T_pD_i} = \frac{\log\left(\frac{N}{df_{T_p}}\right)}{\log(N)} \times \frac{\log\left(\sum_{j=1}^{tf_{T_p,D_i}} \max_{Tag_x \in TAG_{T_p,j}} \{Weight(Tag_x)\} + 0.5\right)}{\log\left(\max_{T_q \in D_i} \{\sum_{j=1}^{tf_{T_q,D_i}} \max_{Tag_x \in TAG_{T_q,j}} \{Weight(Tag_x)\}\} + 1.0\right)} \quad (5)$$

where,

$tf_{T_pD_i}$ is the term frequency of term T_p in document D_i,

$TAG_{T_p,j}$ is the set of markup tags annotated to the $j-th$ occurrence of the term T_p in document D_i,

N is the total number of documents in a collection, and

df_{T_p} is the number of documents in a collection in which term T_p occurs.

In the experiments of [12], use of a rough division of tags outperformed use of a detailed division because we cannot definitely discriminate tags according to their expression power and also because authors of Web pages cannot use tags with seriously considering their expression power. k-NN classifier using only tag's importance did not improve the performance, but k-NN classifier using both tag's importance and feature selection achieved 14.7% improvement of micro-averaging breakeven point. Our observation is that the bad performance of k-NN classifier using only tag's important has been caused by the emphasis of noise terms as well as features [12].

5 New Similarity Measure

k-NN classifier directly uses the k-nearest training documents with respect to a test document to calculate the likelihood of categories. The most important part in k-NN classifier is document-document similarity measure used in the selection of k-nearest neighbors. Most previous k-NN classifiers use the cosine similarity measure [2, 10, 11, 12, 16, 17, 18, 19]. The similarity measures used in vector space model like cosine coefficient and inner product have a weak point that the measures cannot take the advantage of the association between terms [8]. The association between terms constrains each other's semantic concept. Because of the weakness of vector space, the k-nearest neighbors to a test document are frequently selected, depending on only one or a few terms having high weight in the test document. For example, three-dimensional term space <*bank, deposit, river*> identifies each document and the each term is assigned with the weight as shown in (*a*) of Figure 1. From (*a*) of Figure 1, we know that document D_1 is more similar with document D_2 than document D_3. However, both of the inner product and the cosine measures produce the reverse results of our intuition, as shown in (*b*) of Figure 1, because of the weight of the term *bank*. Although terms with low weight are not important in a document, the minor terms can constrain the semantic concept of major terms in the document.

	bank	*deposit*	*river*
D	0.8	0.2	0
D	0.2	0.8	0
D	0.8	0	0.2

(*a*) term-document matrix

	inner product	cosine coefficient
$sim(D, D)$	0.32	0.47
$sim(D, D)$	0.64	0.94
$sim(D, D)$	0.16	0.10

(*b*) similarities between two documents

Figure 1. An example for the similarities in the term vector space

The terms in a document might share a certain relationship to describe the topic of the document. If many terms in the document co-occur in another document, these matching terms in the second document also share the same relationship as they share in the first document. The more the number of the matching terms are, the stronger the coherence of the shared relationship between two documents is. The number of the matching terms between two documents is related to the information of terms association extracted from the inside resource (two documents). We call the number of the matching terms the "matching factor". In this section, we propose a similarity measure that is modified from a traditional similarity measure to use the matching factor as follows:

$$sim\_mf(D_x, D_i) = \left(a + \frac{mf}{|D_x|} + \frac{mf}{|D_i|}\right)^{mf-1} \times sim(D_x, D_i) \quad (6)$$

where

mf is the number of the matching terms between document D_x and document D_i,

a is a constant for the importance of matching factor (default : 1.2), and

$|D_i|$ is the total number of terms in document D_i.

In Formula 6, $sim(D_x, D_i)$ is the similarity of a traditional measure like cosine and inner product measure. The modified cosine similarity with matching factor between document D and document D in Figure 1 is 1.19 and the modified inner product similarity with matching factor between document D and document D is 0.81. The other similarities in (*b*) of Figure 1 are not changed because the matching factor mf is 1. Hence, our proposed similarity measure using matching factor is reasonable in the example.

6 Test Collection

For an evaluation of Web page classification, we chose a test collection, which consists of 32,442 Web pages and 8,702 categories. Web pages and categories of the test collection are service by *HanMir* at October 20, 1999. *HanMir* is one of the most popular search engines in Korea. The categories of the test collection have an 11-level hierarchy and are not domain-specific. Categories in our test collection are organized in a hierarchy of increasing specificity. To apply a standard classification approach to Web page classification, we simply constructed a flattened class space with one class for every category having Web pages in the hierarchy. 7,700 categories among 8,702 categories have the Web pages. The research dealing with the relationship of the hierarchy is beyond this study, so that we ignored the hierarchy in this paper. Table 1 shows the distribution of categories according to the number of Web pages assigned to each category. As Table 1 shows, 96.36% of total categories appear in less than 25 Web pages, because users desire 5 ~ 20 Web pages per category, so to easily search relevant Web pages to their information need. The most common category appears in only 182 Web pages of which the proportion to total Web pages is only 0.56%. In the test collection, the chance of the most common category assignment being correct is 0.56% (182/32442×100). This precision is very low compared with about 30% in Reuter-21578 collection, widely used in text categorization tasks [5, 6, 12, 18]. The average number of categories per Web site is 1.39. Therefore, the chance for a random assignment being correct is 0.018% (1.3/7700×100).

Table 1. The distribution of categories according to the number of Web sites assigned with each category

the number of Web sites assigned with each category	the number of category
1 ~ 5	5,558 (72.18%)
6 ~ 10	1,124 (14.40%)
11 ~ 25	753 (9.78%)
26 ~ 50	187 (2.43%)
51 ~ 150	75 (0.97%)
150 ~ 182	3 (0.04%)

We arbitrarily split the Web pages into a training set (29,925 Web pages) and a test set (2,517 Web pages). There are no overlap between the training Web pages and the test Web pages. The testing set has 2,874 categories and the training set has 7,583 categories. The average number of categories per a Web page in the training set is 1.36 and the average number of categories per a Web page in the test set is 1.71.

In our test collection, there are 32,442 scripts that explain about every Web page and are composed by human beings. Figure 2 shows the script for the HanMir Web site. The script is originally written in Korean, but we translated the Korean description to English for convenience. The category "Enterprise, Company | Industrial Classification | Communication | Korea Telecom | Service | HanMir" is 6 level category in the category hierarchy. In this category, the character "|" describes the boundary between two levels. So that, "Enterprise, Company" is 1 level category, "Enterprise, Company | Industrial Classification" is 2 level category, and so on. All scripts consist of less than 50 Korean words or English words.

<id> 10235 </id>

<category> "Enterprise, Company | Industrial Classification | Communication | Korea Telecom | Service | HanMir" </category>

<url> http://www.hanmir.com/ </url>

<title> HanMir </title>

<script> HanMir is a portal site that is serviced by Korea Telecom. In the site, Web search engine and telephone number search engine are serviced. And also, the site provides free E-mail service, Japanese-Korean translation, multimedia surfing, Usenet service and so on. </script>

Figure 2. An example of script about HanMir Web site

To collect the training samples, we gathered Web pages in the training set through the Internet. Unfortunately, 2,712 Web pages were eliminated, not connected by the Internet or moved to other hosts. And also, 2,323 Web pages among gathered training samples have no term after the removal of stop-words because these Web pages consist of a brief greeting sentence, image, java script, flash, and other non-textual information. Hence, we could not use the Web pages as training samples. In the situation, the scripts for Web pages that consist of keywords extracted by domain experts are more appropriate for training samples than home pages, which are likely to have noise terms such as typographical errors and advertising descriptions. In our test collection, the training samples are the scripts of Web pages in the training set and the test documents are Web pages gathered through the Internet.

7 Experiments and Results

7.1 Experiments Set Up

In this section, we evaluated the performance of the Web page classifier based on proposed k-NN classifier. To optimize the parameter k for k-NN approach and thresholds of MI and EMI for feature selection, we split the training samples into a training set (90% scripts of original training set) and a test set (10% home pages of original training set) for a validation test. In the validation test, we used an original cosine similarity measure as the similarity measure to find k nearest neighbors to a test Web page. We firstly represented the training samples of validation collection as features selected by 4 cases, according to thresholds of MI and EMI as follows: (1) MI = $-\infty$, EMI = $-\infty$ (no use of feature selection), (2) MI = 2.0, EMI = 0.00025, (3) MI = 2.0, EMI = 0.0003, and (4) MI = 2.0, EMI = 0.00035. Next, for each of the parameter k = 7, 9, 11, 15, 17, 21, 31, 41, 51, 101, 151, we evaluated the efficiency of each k on different feature-set sizes as micro-averaging breakeven point. We chose the parameter k = 31, threshold of MI = 2.0, and threshold of EMI = 0.00025 with the best performance. In this test, k-NN classifier using feature selection (micro-averaging breakeven point of 18.32%) improved the performance of micro-averaging breakeven point by 14.9% in contrast to k-NN classifier not using feature selection (micro-averaging breakeven point of 15.94%). The size of feature set selected by using the thresholds in original training samples is 64,878. To represent test Web pages using tag's importance, we divided tags into three groups and assigned the weight to each group with best results in [12] as shown in Table 2. Hereafter, we used the parameter, thresholds and tag's weight as defined in the validation test.

Table 2. The tags and user-defined weights for each group with best results in [12]

division	weight	markup tags
group 1	4	<title>, <h1>, <meta keyword>, <meta description>
group 2	3	<b>, <blink>, <h2-7>, , <u>, <i>, <big>, <dt>, <dfn>, <caption>, <abstract>, , <alt>, <a>, <strike>, <note>, <q>, <footnote>, <cite>, , , <option>, <role>
group 3	1	the others

7.2 Results

To evaluate the document-document similarity measure using a matching factor, we compared four similarity measures: cosine similarity, inner product, cosine similarity using matching factor, and inner product using matching factor. The results of the comparative experiments are shown in Table 3. The results indicate that the matching factor efficiently finds relevant k-nearest neighbors to a test document in term vector space.

Table 3 gives a very low performance in contrast to the best performance of previous Web page categorization in the Yahoo collection [1, 9]. The Yahoo collection is constructed by dividing Web pages pointed by the *Yahoo!* 'Science' hierarchy in July 1997 into topics by chopping

Table 3. The results of the experiments for the comparison between the document-document similarity measures

similarity measure	micro-averaging breakeven point (%)
cosine similarity	18.23
cosine similarity using matching factor	19.23 (+5.5%)
inner product	19.74
inner product using matching factor	20.02 (+1.4%)

the hierarchy two levels deep. There are two different versions for the Yahoo collection, according to the number of categories and Web pages. In [1], a naive Bayes approach achieved the micro-averaging breakeven point 54% in a Yahoo collection that consists of 13,589 pages and 95 categories. In [9], a naive Bayes approach achieved the micro-averaging breakeven point 66.4% in a Yahoo collection that consists of 6,294 pages and 41 categories. Unlike Yahoo collection, the levels of every category in our collection are not the same, so that many categories are conceptually overlapped. Categories and Web pages in our test collection are not domain-specific, so the meanings of terms are more ambiguous in our test collection than the Yahoo collection. Furthermore, training samples are not sufficient to learn classifiers for each category correctly in our test collection. As compared with the Yahoo collection, such a low performance can be explained by such negative factors in our test collection.

To increase the number of training samples for each category, we cut off the hierarchy in each level and next combined training samples and test documents of each chopped category into training samples and test documents of its nearest ancestor among the remainders respectively.

Table 4. The results of the experiments at each level chopped

level	the number of used categories	micro-averaging breakeven point (%)
1	16	59.64%
2	270	50.31%
3	1,529	37.66%
4	3,484	34.96%
5	5,007	25.56%
6	5,647	24.15%
7	6,051	22.55%
8	7,090	20.89%
9	7,438	20.85%
10	7,694	20.02%
11	7,700	20.02%

Table 4 shows the number of used categories and the performance of k-NN approach using inner product with matching factor at each revised collection. Because any category of one level has not training samples in original test collection, the collection by chopping two levels has 270 categories in the same level. The collection among the revised collections is the most similar with Yahoo collection. By indirectly comparing the performance of our k-NN approach on the collection with the performance of the other approaches proposed in [1, 9] tested on the Yahoo collection, we determine that our approach for Web page classification performs favorably in spite of the remained differences between two collections such as languages, domain, and the number of categories.

In order to help users to easily find the Web pages relevant to their needs through several paths of the hierarchy, the hierarchy includes many categories that have two or more parents linked from different roots. And also, every relationship between parents and children is not "is-a" relation in the hierarchy. Because of the ill-defined relationship of the hierarchy, many training samples of a category combined with training samples of its descendants might resemble the training samples of others than its training samples. This phenomenon reaches its climax when we construct the collection by chopping one level. This reason caused that the performance on the collection by chopping one level as shown in Table 4 falls short of our expectations, although the collection has only 16 categories.

As mentioned in Section 6, categories used in the experiments have not the same level in the hierarchy. Our Web page classifier often predicted the descendents of correct categories attached by human. In the former evaluation, we considered the case to be incorrect prediction. But, such predictions can help domain experts to classify Web pages. We presumed that the categories are correct and then evaluated our Web page classifier. Web page classifier achieved the micro-averaging breakeven point of 26.21%.

8 Conclusion

In this paper, we proposed a Web page classifier based on an adaptation of k-NN approach. The k-NN approach is characterized as follows: a feature selection method to reduce noise terms in training samples, a term weighting scheme using HTML tags, and a document-document similarity measure with matching factor proposed in this paper. In our experiments, the similarity measure proved successful to improve the efficiency of k-NN approach. And also, the use of feature selection in k-NN approach improved the performance in the validation test.

In future work, we will investigate the classification method that uses the linkage information of a Web page to classify Web sites or pages using the Web pages linked by new Web page (home page). And also, we will investigate the method that increases training samples using

connectivity analysis in the WWW and then evaluate the proposed classifiers using the extended training samples. We will also investigate the classification method that can deal with the relationship of categories in an existing hierarchy such as commercial Web directories.

References

1. Andrew McCallum and Kamal Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," In AAAI-98 Workshop on Learning for Text Categorization, 1998. http://www.cs.cmu.edu/~mccallum.
2. Brij Masand, Gordon Linoff and David Waltz, "Classifying News Stories using Memory Based Reasoning," In Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval (SIGIR'92), pp. 59-65, Copenhagen, Denmark, 1992.
3. Chidanand Apté and Fred Damerau, "Automated Learning of Decision Rules for Text Categorization," ACM Transactions on Information Systems, Vol. 12, No. 3, pp. 233-251, 1994.
4. C. J. Van Rijsbergen, "A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval," Journal of Documentation, Vol. 33, No. 2, pp. 106-119, June 1977.
5. David D. Lewis, "Representation and Learning in Information Retrieval," PhD thesis, Department of Computer Science; Univ. of Massachusetts; Amherst, MA 01003, 1992.
6. David D. Lewis, "An Evaluation of Phrasal and Clustered Representations on a Task", In Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval (SIGIR'92), pp. 37-50, Copenhagen, Denmark, 1992.
7. David D. Lewis and Marc Ringuette, "A comparison of two learning algorithms for text categorization," In Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94), University of Nevada, Las Vegas, USA, pp. 81-93, 1994.
8. G. Salton, "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer," Addison-Weseley, Reading, Massachusetts, 1989.
9. L. Douglas Baker and Andrew Kachites McCallum, "Distributional Clustering of Word for Text Classification," In Proceedings of the 21th Annual International Conference on Research and Development in Information Retrieval (SIGIR'98), Melbourne, Australia, pp. 96-103, 1998.
10. Leah Larkey and W. Bruce Croft, "Combining Classifiers in Text Categorization," In Proceedings of the 19th International Conference on Research and Development in Information Retrieval (SIGIR '96), pp. 289-297, Zurich, Switzerland, 1996
11. Makoto Iwayama and Takenobu Tokunaga, "Cluster-Based Text Categorization: A Comparison of Category Search Strategies," In Proceedings of the 18th Annual International Conference on Research and Development in Information Retrieval (SIGIR'95), pp. 273-280, Seattle, Washington, USA, 1995.
12. Oh-Woog Kwon, Sung-Hwa Jung, Jong-Hyeok Lee, and Geunbae Lee, "Evaluation of Category Features and Text Structural Information on a Text Categorization Using Memory Based Reasoning," In Proceedings of the 18th International Conference on Computer Processing of Oriental Languages (ICCPOL'99), pp. 153-158, University of Tokushim, Japan, 1999.
13. Susan Dumais, John Platt, David Heckerman, and Mehran Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," In Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM'98), 1998, http://robotics.stanford.edu/users/sahami/papers.html.
14. Thorsten Joachims, "Text Categorization with Support Vector Machine: Learning with Many Relevant Feature," In Proceedings of European Conference on Machine Learning (ECML), 1998, http://www-ai.cs.uni-dortmund.de/PERSONAL/joachims.eng.html.
15. Thorsten Joachims, "Transductive Inference for Text Classification using Support Vector Machines," In Proceedings of International Conference on Machine Learning (ICML), 1999, http://www-ai.cs.uni-dortmund.de/PERSONAL/joachims.eng.html.
16. Yiming Yang, "Expert Network: Effective and Efficient Learning from Human Decision in Text Categorization and Retrieval," In Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval (SIGIR'94), pp. 13-22, Dublin, Ireland, 1994.
17. Yiming Yang, "An Evaluation of Statistical Approach to Text Categorization," Information Retrieval, Vol. 1, No. 1/2, pp. 69-90, 1999.
18. Yiming Yang, and Xin Lui, "A Re-examination of Text Categorization Methods," In Proceedings of the 22th Annual International Conference on Research and Development in Information Retrieval (SIGIR'99), pp. 42-49, University of California, Berkeley, USA, 1999.
19. Wai Lam and Chao Yang Ho, "Using A Generalized Instance Set for Automatic Text Categorization," In Proceedings of the 21th Annual International Conference on Research and Development in Information Retrieval (SIGIR'98), Melbourne, Australia, pp. 81-89, 1998.