Multi-Instance Learning Based Web Mining

Zhi-Hua Zhou*, Kai Jiang, and Ming Li

National Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China

Abstract

In *multi-instance learning*, the training set comprises labeled *bags* that are composed of unlabeled instances, and the task is to predict the labels of unseen bags. In this paper, a web mining problem, i.e. web index recommendation, is investigated from a multi-instance view. In detail, each web index page is regarded as a bag, while each of its linked pages is regarded as an instance. A user favoring an index page means that he or she is interested in at least one page linked by the index. Based on the browsing history of the user, recommendation could be provided for unseen index pages. An algorithm named Fretcit-*k*NN, which employs the Minimal Hausdorff distance between frequent term sets and utilizes both the *references* and *citers* of an unseen bag in determining its label, is proposed to solve the problem. Experiments show that in average the recommendation accuracy of Fretcit-*k*NN is 81.0% with 71.7% recall and 70.9% precision, which is significantly better than the best algorithm that does not consider the specific characteristics of multi-instance learning, whose performance is 76.3% accuracy with 63.4% recall and 66.1% precision.

Key words: Machine Learning; Data Mining; Multi-Instance Learning; Web Mining; Web Index Recommendation; Text Categorization

* Corresponding author. Tel.: +86-25-359-3163; fax: +86-25-330-0710; E-mail address: zhouzh@nju.edu.cn

1 Introduction

At present, roughly speaking, there are three frameworks for *learning from examples* [14]. That is, supervised learning, unsupervised learning, and reinforcement learning. Supervised learning attempts to learn a concept for correctly labeling unseen examples, where the training examples are with labels. Unsupervised learning attempts to learn the structure of the underlying sources of examples, where the training examples are with no labels. Reinforcement learning attempts to learn a mapping from states to actions, where the examples are with no labels but with delayed rewards that could be viewed as delayed labels.

In investigating the problem of drug activity prediction, Dietterich et al. [10] proposed the notion of *multi-instance learning*, where the training set is composed of many *bags* each containing many instances. A bag is positively labeled if it contains at least one positive instance. Otherwise it is labeled as a negative bag. The task is to learn some concept from the training set for correctly labeling unseen bags. Such a task is quite difficult because although the labels of the training bags are known, that of the training instances are not available. It has been shown that learning algorithms ignoring the characteristics of multi-instance problems, such as popular decision trees and neural networks, could not work well in this scenario [10].

Since multi-instance problems extensively exist but they are unique to those addressed by previous learning frameworks, multi-instance learning is regarded as a new learning framework [14], and has attracted much attention of the machine learning community.

In this paper, a specific web mining task, i.e. recommending web index pages based on user behavior, is studied. Experiments show that when the problem is regarded as a traditional supervised learning problem, the performance of the best learning algorithm investigated is about 76.3% accuracy with 63.4% recall and 66.1% precision. However, if this problem is regarded as a multi-instance problem, significantly better solution could be obtained. In fact, this paper proposes a

multi-instance learning algorithm named Fretcit-kNN, i.e. FREquent Terms based CITation-kNN, to solve the web index recommendation problem and achieves about 81.0% accuracy with 71.7% recall and 70.9% precision.

The rest of this paper is organized as follows. Section 2 introduces multi-instance learning. Section 3 describes the problem of web index recommendation. Section 4 presents Fretcit-kNN and its variant. Section 5 reports the experiments. Finally, Section 6 summarizes the main contributions of this paper and raises several issues for future work.

2 Multi-Instance Learning

In the middle of 1990s, Dietterich et al. [10] investigated the problem of drug activity prediction. The goal was to endow learning systems with the ability of predicting that whether a new molecule was qualified to make some drug, through analyzing a collection of known molecules.

Most drugs are small molecules working by binding to larger protein molecules such as enzymes and cell-surface receptors. For molecules qualified to make a drug, one of its low-energy shapes could tightly bind to the target area. While for molecules unqualified to make a drug, none of its low-energy shapes could tightly bind to the target area. The main difficulty of drug activity prediction lies in that each molecule may have many alternative low-energy shapes, but biochemists only know that whether a molecule is qualified to make a drug or not, instead of knowing that which of its alternative low-energy shapes responses for the qualification.

An intuitive solution is to utilize supervised learning algorithms by regarding all the low-energy shapes of the 'good' molecules as positive training examples, while regarding all the low-energy shapes of the 'bad' molecules as negative training examples. However, as shown by Dietterich et al. [10], such a method can hardly work due to high false positive noise, which is caused by that a 'good' molecule may have hundreds of low-energy shapes but maybe only one of them is really a

'good' shape. In order to solve this problem, Dietterich et al. [10] regarded each molecule as a bag, and regarded the alternative low-energy shapes of the molecule as the instances in the bag, thereby formulated multi-instance learning.

The PAC-learnability of multi-instance learning has been studied by many researchers. Long and Tan [13] showed that if the instances in the bags are independently drawn from product distribution, then the APR (Axis-Parallel Rectangle) proposed by Dietterich et al. [10] is PAC-learnable. Auer et al. [4] showed that if the instances in the bags are not independent then APR learning under the multi-instance learning framework is NP-hard. Moreover, they presented a theoretical algorithm that does not require product distribution, which was transformed to a practical algorithm named MULTINST later [3]. Blum and Kalai [5] described a reduction from PAC-learning under the multi-instance learning framework to PAC-learning with one-sided random classification noise. They also presented a theoretical algorithm with smaller sample complexity than that of Auer et al.'s algorithm [4].

A representative practical multi-instance learning algorithm is Diverse Density proposed by Maron and Lozano-Pérez [15]. Intuitively, *diverse density* at a point in the feature space is defined to be a measure of how many different positive bags have instances near that point, and how far the negative instances are from that point. Thus, the task of multi-instance learning is transformed to search for a point in the feature space with the maximum diverse density. The Diverse Density algorithm has been applied to several applications including natural scene classification [16], stock selection [15], and content-based image retrieval [20].

There are also many other practical multi-instance learning algorithms, such as Wang and Zucker's extended k-nearest neighbor algorithms [19], Ruffo's multi-instance decision tree Relic [18], Chevaleyre and Zucker's multi-instance decision tree ID3-MI and rule inducer RIPPER-MI [6], Zhou and Zhang's multi-instance neural network BP-MIP [23], and Zhang and Goldman's EM-DD [21]. The EM-DD algorithm has been applied to content-based image retrieval [22]. Recently, Zhou and Zhang obtained the best result up to now on the benchmark test of

multi-instance learning with EM-DD ensembles [24].

In the early years of the research of multi-instance learning, most work were on multi-instance classification with discrete-valued outputs. Recently, multi-instance regression with real-valued outputs begins to attract the attention of some researchers. Ray and Page [17] showed that the general formulation of the multi-instance regression task is NP-hard, and proposed an EM-based multi-instance regression algorithm. Amar et al. [2] extended the Diverse Density algorithm for multi-instance regression, and designed some method for artificially generating multi-regression data.

It is worth mentioning that multi-instance learning has even attracted the attention of the ILP community. De Raedt [8] showed that multi-instance problems could be regarded as a bias on inductive logic programming. He also suggested that the multi-instance paradigm could be the key between the propositional and relational representations, being more expressive than the former, and much easier to learn than the latter. Zucker and Ganascia [25][26] presented REPEAT, an ILP system based on an ingenious bias which firstly reformulated the relational examples in a multi-instance database, and then induced the final hypothesis with a multi-instance learner.

3 Web Index Recommendation

There are diverse web pages on the Internet, among which some pages contain plentiful information but themselves only provide titles or brief summaries while leaving the detailed presentation to their linked pages. These web pages are called *web index pages*. For example, the entrance of NBA at Yahoo! (sports.yahoo.com/nba/) is a web index page.

Everyday a web user may encounter many web index pages. Some of these pages may contain issues interested the user while some may not. It is nice if these pages could be automatically analyzed so that only the index pages containing interesting

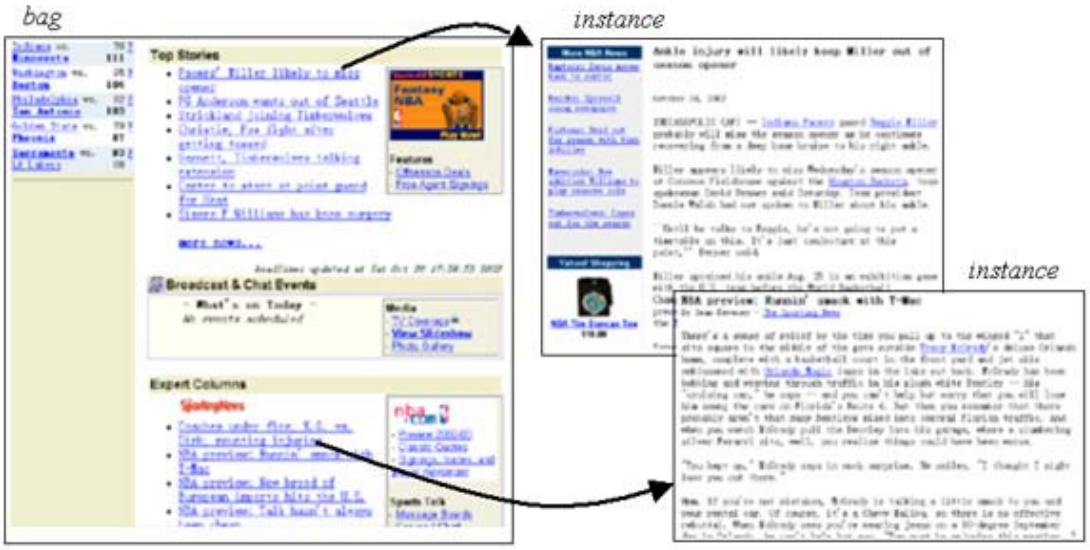

Fig. 1. The web index page is regarded as a bag, while its linked pages are regarded as the instances in the bag

issues are presented to the user. That is, through analyzing the web index pages that the user has browsed, try to identify whether a new web index page will interest the user or not. This problem is called *web index recommendation*, which is a specific web usage mining task, and the solution to the problem may be helpful for developing user-adaptive intelligent web browsers. Here the difficulty lies in that the user only specifies whether he or she is interested in an index page, instead of specifying the concrete links that he or she is really interested in.

This problem could be viewed as a multi-instance problem. Now the goal is to label unseen web index pages as positive or negative. A positive web index page is such a page that the user is interested in at least one of its linked pages. A negative web index page is such a page that none of its linked pages interested the user. Thus, each index page could be regarded as a bag while its linked pages could be regarded as the instances in the bag. For illustration, Fig. 1 shows a bag and two of its instances.

For simplifying the analysis, this paper only focuses on the hypertext information on the pages while neglecting other hypermedia such as images, audios, videos, etc. Then, each instance can be represented by a term vector $\mathbf{T} = [t_1, t_2, \cdots, t_n]$,

where t_i $(i = 1, 2, \cdots, n)$ is one of the n most frequent terms appearing in the corresponding linked page. $\mathbf{T}$ could be obtained by pre-accessing the linked page and then counting the occurrence of different terms. Note that some trivial terms such as '*a*', '*the*', '*is*', are neglected in this process. In this paper, all the pages are described by the same number of frequent terms, i.e. the length of any term vectors are the same. However, for term vectors corresponding to different instances, even though their length is the same, their components may be quite different. Moreover, for different bags, since their corresponding web index pages may contain different number of links, the number of instances in the bags may be different.

Thus, a web index page linking to m pages, i.e. a bag containing m instances, can be represented as $\{[t_{11}, t_{12}, \cdots, t_{1n}], [t_{21}, t_{22}, \cdots, t_{2n}], \cdots, [t_{m1}, t_{m2}, \cdots, t_{mn}]\}$. The label of the bag is positive if the web index page interested the user. Otherwise the label is negative.

Note that the web index pages may contain many links to advertisements or other index pages, which may baffle the analysis. In this paper it is constrained that for a linked page to be considered as an instance in a bag, its corresponding link in the index page must contain at least four terms. It is surprising that such a simple strategy helps remove most useless links.

4 Fretcit-kNN and Its Variant

kNN, i.e. k-nearest neighbor [7], is a popular lazy learning algorithm [1], which labels an unseen example with the label holding by majority of its k nearest neighboring training examples. Dietterich et al. [9] have shown that standard kNN with Euclidean distance or Tangent distance could hardly be used to solve the drug discovery problem. In order to adapt kNN to multi-instance problems, Wang and Zucker [19] employed modified Hausdorff distance [11] to measure the neighboring distances between numerical objects.

By definition, two sets A and B are within Hausdorff distance d of each other

if and only if every object of A is within distance d of at least one object of B, and every object of B is within distance d of at least one object of A. Formally, the Hausdorff distance between A and B is defined as Eq. 1, where $\|a-b\|$ is the Euclidean distance between a and b.

$$\mathrm{H}(A,B) = \mathrm{Max}\{\underset{a\in A}{\mathrm{Max}}\,\underset{b\in B}{\mathrm{Min}}\,\|a-b\|, \underset{b\in B}{\mathrm{Max}}\,\underset{a\in A}{\mathrm{Min}}\,\|b-a\|\} \tag{1}$$

Since the standard Hausdorff distance is very sensitive to outliers, Wang and Zucker [19] defined the Minimal Hausdorff distance as Eq. 2.

$$\mathrm{minH}(A,B) = \underset{a\in A,\ b\in B}{\mathrm{Min}}\,\|a-b\| \tag{2}$$

Wang and Zucker [19] also suggested taking into account not only the neighbors of a bag A, i.e. A's *references*, but also the bags that count A as a neighbor, i.e. A's *citers*. The R-nearest references of bag A are defined as the R-nearest neighbors of A. The C-nearest citers of A are defined as the bags that regard A as their C-nearest neighbors. The label of an unseen bag is determined by majority voting among its R-nearest references and C-nearest citers. If tie appears, the bag is labeled as negative.

It is obvious that since both the standard Hausdorff distance and the Minimal Hausdorff distance employ Euclidean distance, they can only be applied to numerical objects, i.e. instances described by only numerical attributes. However, in the problem of web index recommendation, the instances are described by unordered attributes whose possible values are textual frequent terms. Therefore new distance measure must be developed for the web index recommendation problem.

For two sets of frequent terms A and B each containing n terms, i.e. $\{a_1, a_2, \cdots, a_n\}$ and $\{b_1, b_2, \cdots, b_n\}$, an intuitive way to measure their similarity is to count the number of shared terms. The more the shared terms, the bigger the similarity. For example, suppose $A = \{red, white, yellow\}$, $B = \{black, red, yellow\}$, and $C = \{gray, green, yellow\}$. It is obvious that A is more similar to B than to C because the number of terms shared by A and B, i.e. 2, is bigger than that shared by A and C, i.e. 1.

Based on this heuristics, the Minimal Hausdorff distance between frequent term sets can be defined as Eq. 3.

$$\text{fret-minH}\,(A,B) = \underset{a\in A,\ b\in B}{\text{Min}}\left(1 - \sum_{\substack{i,j=1\\ a_i=b_j}}^{n} \frac{1}{n}\right) \tag{3}$$

Through employing fret-minH(.) to measure the neighboring distance between bags, and utilizing both the references and the citers of an unseen bag in determining its label, the Fretcit-kNN algorithm is obtained. It is evident that such an algorithm can be applied to the problem of web index page recommendation.

During counting the occurrence of different terms in the linked pages, the frequency of the terms can be obtained, which can be used to rank the frequent terms. This rank information could also be utilized in measuring the distances between bags.

For example, suppose $A = \{red, white, yellow\}$, $B = \{black, red, pink\}$, and $C = \{gray, green, red\}$, where the terms have been ranked according to the descending order of their frequencies. If fret-minH(.) is used to measure the distance, then B and C are equally close to A because each of them shares one term with A. However, from intuition, A should be more similar to B than to C because the shared term, i.e. *red*, is the 2nd frequent term of B while is only the 3rd frequent term of C.

Based on this heuristics, the Minimal Hausdorff distance between ranked frequent term sets can be defined as Eq. 4, where Rank(.) is a function returning the normalized rank of a term. For example, Rank(*red*) is $0.53(= 2/(3^2 + 2^2 + 1^2)^{1/2})$ for $\{black, red, pink\}$ while is $0.27(= 1/(3^2 + 2^2 + 1^2)^{1/2})$ for $\{gray, green, red\}$. If several terms are with the same frequency, then the average rank are used. For example, suppose the frequency of *black* and *red* are the same for $\{black, red, pink\}$, then Rank(*black*) = Rank (*red*) $= 0.68(= [(3^2 + 2^2) \times 0.5/(3^2 + 2^2 + 1^2)]^{1/2})$.

$$\text{r-fret-minH}\,(A,B) = \underset{a\in A,\ b\in B}{\text{Min}}\left(1 - \sum_{\substack{i,j=1\\ a_i=b_j}}^{n} \text{Rank}\,(a_i)\,\text{Rank}\,(b_j)\right) \tag{4}$$

Through replacing fret-minH(.) with r-fret-minH(.), a variant of Fretcit-kNN, i.e. r-Fretcit-kNN, is obtained. It is obvious that such an algorithm can also be applied to the problem of web index page recommendation.

5 Experiments

5.1 Data and Methodology

113 web index pages are collected and then labeled by nine volunteers according to their interests, which results in nine experimental data sets. Note that since every index page may have lots of linked pages, the data sets are rather big. In fact there are 3,423 linked pages in total, and the volume for storage is 126Mb (30.9Mb after compression). For each data set, 75 web index pages are randomly selected as training bags while the remaining 38 index pages are used as test bags. The number of positive and negative bags in the data sets is tabulated in Table 1.

Table 1
Experimental data sets.

Data set	Training set		Test set	
	Positive	Negative	Positive	Negative
V1	17	58	4	34
V2	18	57	3	35
V3	14	61	7	31
V4	56	19	33	5
V5	62	13	27	11
V6	60	15	29	9
V7	39	36	16	22
V8	35	40	20	18
V9	37	38	18	20

The bags are composed of different number of instances. The biggest bag comprises 200 instances, while the smallest one comprises only 4 instances. In average, each bag contains 30.29 (3,423/113) instances. The data sets are publicly available at http://cs.nju.edu.cn/people/zhouzh/zhouzh.files/publication/annex/milweb-data.rar.

In the experiments, the *accuracy* of recommendation is measured. Moreover, since the data sets contain different number of positive and negative bags, the *recall* and *precision* of the recommendation are also measured.

Suppose there are P positive bags and N negative bags in the test set, among which P_a positive bags are recommended while P_r positive bags are rejected, and N_a negative bags are recommended while N_r negative bags are rejected. It is obvious that $P = P_a + P_r$, $N = N_a + N_r$. Then, the accuracy, recall, and precision are defined as Eqs. 5 to 7, respectively.

$$accuracy = \frac{P_a + N_r}{P + N} \tag{5}$$

$$recall = \frac{P_a}{P} \tag{6}$$

$$precision = \frac{P_a}{P_a + N_a} \tag{7}$$

Moreover, for the convenience of the presentation of the experimental results, two other measures, i.e. *error* and *error ratio*, are used. The error is defined as Eq. 8. Suppose the error of algorithms A and B are $error_A$ and $error_B$ respectively, then the error ratio of algorithm A against algorithm B is defined as Eq. 9.

$$error = 1 - accuracy \tag{8}$$

$$ratio_{A/B} = \frac{error_A}{error_B} \tag{9}$$

For each data set, 5, 8, 10, 12, and 15 frequent terms are used to describe the instances, respectively. In order to show the overall performance of an algorithm on different data sets, the *geometrical mean*, i.e. average value across all data sets, of accuracy, recall, and precision are also provided besides the results on each data set.

5.2 Comparing Txt-kNN and Cit-kNN with Fretcit-kNN

At first, experiments are performed to evaluate the performance of Fretcit-kNN on the web index recommendation problem. Since Fretcit-kNN is an extended kNN algorithm that considers the characteristics of multi-instance problems, for comparison, two extended kNN algorithms that do not consider the characteristics of multi-instance problems are also evaluated.

The first compared algorithm is obtained through adapting the standard kNN algorithm to textual objects. Recall that the standard kNN algorithm utilizes Euclidean distance to measure the distance between examples, which disable it be applied to objects described by textual frequent terms. However, if the distance metric is replaced by fret-minH(.), then the modified algorithm can be easily applied to textual objects. Here the modified algorithm is called Txt-kNN.

The main difference between Txt-kNN and Fretcit-kNN is that the latter is a multi-instance learning algorithm while the former is a single-instance learning algorithm, that is, the former algorithm regards all the instances in a bag have the label of that bag. For example, all the instances of a positive bag are regarded as positive instances by Txt-kNN. In prediction, the unseen bag is positively labeled if at least one of its instances are predicted as a positive instance. Another difference between Txt-kNN and Fretcit-kNN is that the former considers only the references of an unseen object in prediction while the latter considers both the references and the citers.

The second compared algorithm is obtained by enabling Txt-kNN consider both the references and the citers of an unseen object in prediction. Note that it is still a single-instance learning algorithm which regards all the instances in a bag have the label of that bag. In order to distinguish it with the multi-instance learning algorithm that was designed for numerical objects, i.e. Citation-kNN [19], here this algorithm is called Cit-kNN.

Moreover, a classical information retrieval technique, i.e. TFIDF [12], is also eval-

uated on the data sets through regarding all the instances in a bag have the label of that bag, which provides a baseline for the comparison. Note that in principle the TFIDF, Txt-kNN and Cit-kNN algorithms can be directly applied to the web index pages while ignoring their linked pages. But since most information of the index pages are delivered by their linked pages, the corresponding performance is very poor, which is not presented in this paper.

In the experiments the number of references and citers in consideration by both Cit-kNN and Fretcit-kNN are set to 2 and 4, respectively, because in Wang and Zucker's work the best performance of Citation-kNN was obtained under such a configuration [19]. The error ratios of Txt-kNN, Cit-kNN and Fretcit-kNN against TFIDF are depicted in Fig. 2, while the detailed experimental results are tabulated in the Appendix.

Fig. 2 and Tables 2 to 6 show that the recommendation quality of Fretcit-kNN is always better than that of the compared algorithms, when 5, 10, 12, or 15 frequent terms are used to describe the instances. When 8 frequent terms are used, the recommendation quality of Fretcit-kNN is still better than that of TFIDF and Txt-kNN, but comparable to that of Cit-kNN. This might because that the influence of the number of frequent terms on the performance of the extended kNN algorithms are not synchronous. In fact, when 8 frequent terms are used, Cit-kNN reaches its best performance while Fretcit-kNN obtains its worst performance, which results in their comparable recommendation quality.

Pairwise two-tailed t-tests on the nine data sets indicate that the recommendation quality of Fretcit-kNN is significantly better than that of the other compared algorithms. In fact, if the geometrical mean of the corresponding 'GM' values in Tables 2 to 6 are computed across all number of frequent terms, it could be found that the overall performance of Fretcit-kNN is 81.0% accuracy with 71.7% recall and 70.9% precision, which is significantly better than the performance of TFIDF (75.0% accuracy with 62.7% recall and 64.2% precision), Txt-kNN (72.4% accuracy with 78.6% recall and 56.1% precision), and Cit-kNN (75.8% accuracy with 61.9% recall and 63.8% precision). It is evident that these results well support the

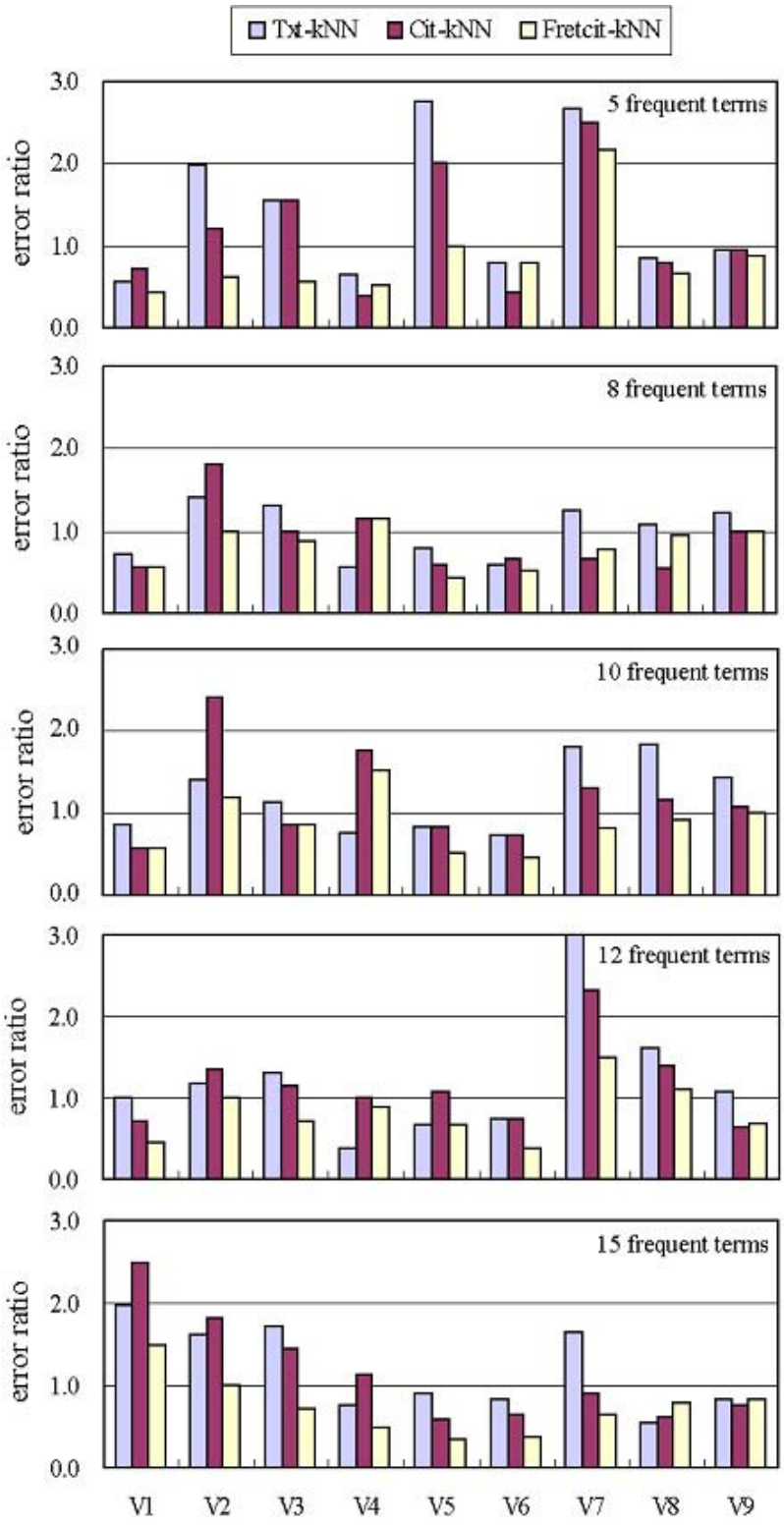

Fig. 2. Comparison of error ratios of Txt-kNN, Cit-kNN and Fretcit-kNN

claim that if the multi-instance nature of the web index recommendation problem is considered, then better solution could be achieved.

Note that the recall of Txt-kNN seems better than that of Fretcit-kNN, which is very misleading. This phenomenon is caused by the fact that the Txt-kNN algorithm has high chances to believe that an unseen bag is a positive bag, because it regards all instances in positive training bags as positive instances, and positively labels an unseen bag as long as it positively labels an unseen instance. In other words, although more positive bags are 'accepted', far more negative bags that should be 'rejected' are also 'accepted'. Thus, the cost of the improvement of recall is the serious deterioration of precision, which leads to its poor accuracy.

It is also worth noting that although Cit-kNN works in a similar way as Txt-kNN does, that is, regards all instances in positive training bags as positive instances and positively labels an unseen bag as long as an unseen instance is positively labeled, its recall is not so misleadingly high and its precision seems have not been sacrificed as that of Txt-kNN. This is because Cit-kNN employs both the references and the citers in labeling an unseen bag, which greatly reduces the chances of wrongly labeling a negative bag as positive. In fact, both the accuracy and precision of Cit-kNN are significantly better than that of Txt-kNN.

5.3 Comparing r-Txt-kNN and r-Cit-kNN with r-Fretcit-kNN

Then, experiments are performed to evaluate the performance of r-Fretcit-kNN on the web index recommendation problem. Since r-Fretcit-kNN is an extended kNN algorithm that considers both the rank information of the frequent terms and the characteristics of multi-instance problems, for comparison, two extended kNN algorithms that consider the rank information of the frequent terms but do not consider the characteristics of multi-instance problems are also evaluated.

The compared algorithms, i.e. r-Txt-kNN and r-Cit-kNN, are obtained by replacing the distance metric used by Txt-kNN and Cit-kNN with r-fret-minH(.). Note that both of them are single-instance learning algorithms, which regards all the instances

in a bag have the label of that bag. The error ratios of r-Txt-kNN, r-Cit-kNN and r-Fretcit-kNN against TFIDF are depicted in Fig. 3, while the detailed experimental results are tabulated in the Appendix.

Fig. 3 and Tables 2 to 6 show that the recommendation quality of r-Fretcit-kNN is always better than that of the compared algorithms, no matter how many frequent terms are used to describe the instances.

Pairwise two-tailed t-tests on the nine data sets indicate that the recommendation quality of r-Fretcit-kNN is significantly better than that of the other compared algorithms. In fact, if the geometrical mean of the corresponding 'GM' values in Tables 2 to 6 are computed across all number of frequent terms, it could be found that the overall performance of r-Fretcit-kNN is 80.4% accuracy with 71.6% recall and 71.5% precision, which is significantly better than the performance of TFIDF (75.0% accuracy with 62.7% recall and 64.2% precision) and r-Txt-kNN (73.6% accuracy with 74.7% recall and 57.0% precision). It is also significantly better than the best single-instance learning algorithm, i.e. r-Cit-kNN, whose performance is 76.3% accuracy with 63.4% recall and 66.1% precision. It is evident that these results well support the claim that if the multi-instance nature of the web index recommendation problem is considered, then better solution could be achieved.

Note that the recall of r-Txt-kNN seems better than that of r-Fretcit-kNN, which is very misleading. The explanation for this phenomenon is the same as that has been discussed in Section 5.2.

5.4 Comparing Fretcit-kNN with r-Fretcit-kNN

Section 5.2 and 5.3 have shown that both Fretcit-kNN and r-Fretcit-kNN are significantly better than their single-instance counterworkers. Then, it is interesting to see which one of them is better for the web index recommendation problem.

In fact, if the geometrical mean of the corresponding 'GM' values of Fretcit-kNN and r-Fretcit-kNN in Tables 2 to 6 are computed across all number of frequent

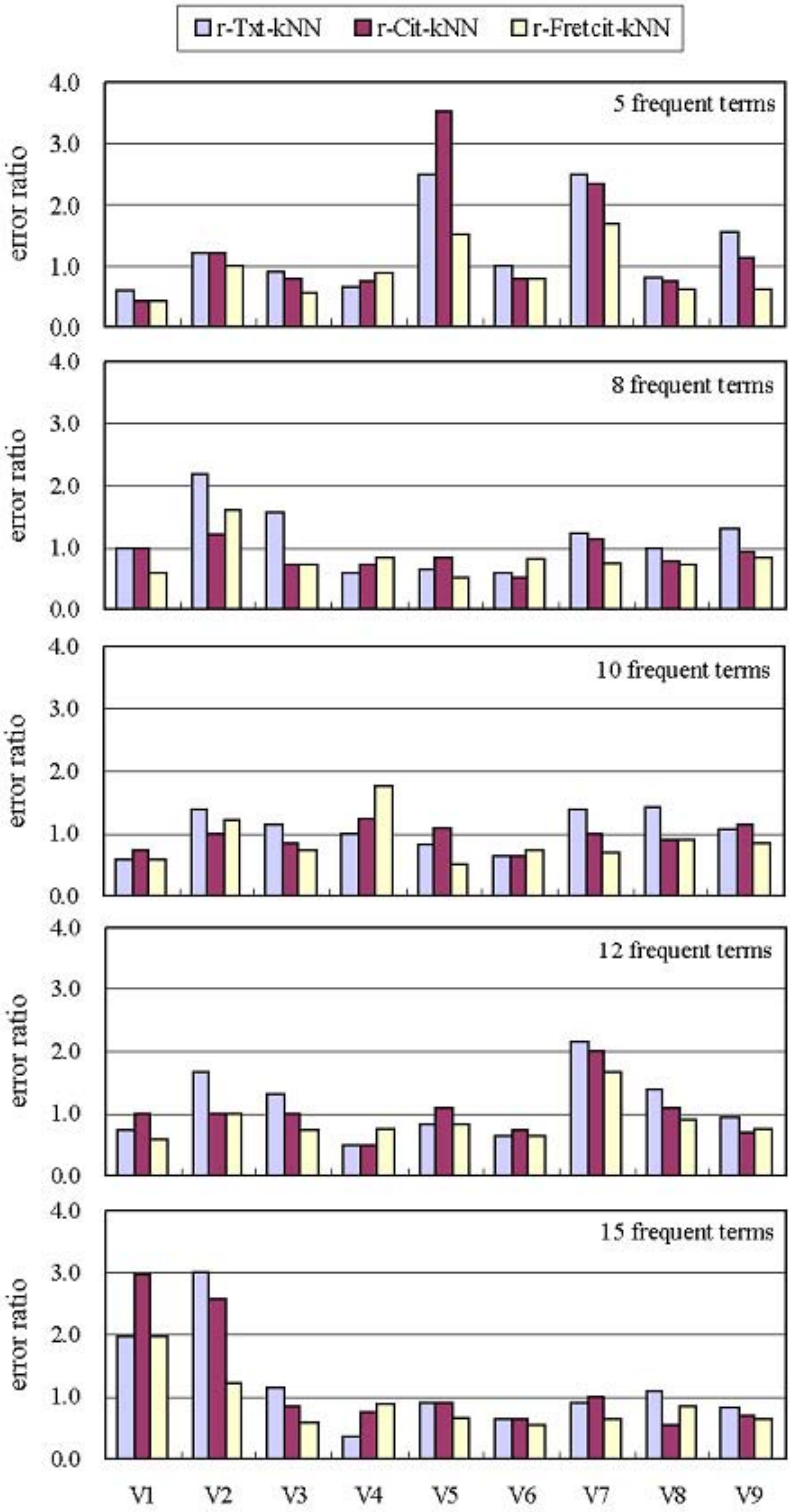

Fig. 3. Comparison of error ratios of r-Txt-kNN, r-Cit-kNN and r-Fretcit-kNN

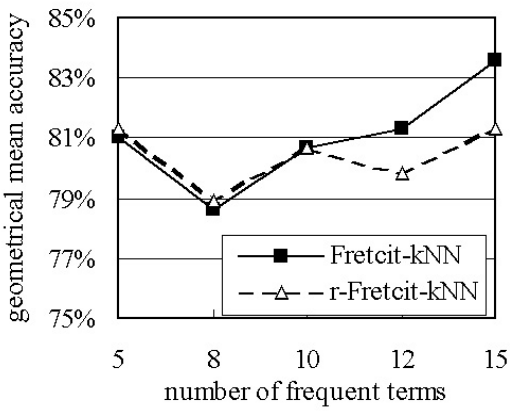

Fig. 4. Comparison of geometrical mean accuracies of Fretcit-kNN and r-Fretcit-kNN

terms, it could be found that the overall performance of Fretcit-kNN is 81.0% accuracy with 71.7% recall and 70.9% precision while that of r-Fretcit-kNN is 80.4% accuracy with 71.6% recall and 71.5% precision. It is obvious that the performance of these two algorithms are very comparable.

The geometrical mean accuracies of Fretcit-kNN and r-Fretcit-kNN under different number of frequent terms that are used to describe the instances are depicted in Fig. 4, which shows that the increase of the number of frequent terms does not necessarily bring improvement because the accuracies of both algorithms deteriorate when the number of frequent terms increases from 5 to 8. This is not strange since for two representations, the better one for learning might not be the longer one.

It is worth noting that r-Fretcit-kNN has utilized more information than Fretcit-kNN does but its performance is not better than that of Fretcit-kNN. In fact, it could be found that although introducing the rank information of the frequent terms has improved Txt-kNN with 1.2% accuracy and Cit-kNN with 0.5% accuracy, it deteriorates Fretcit-kNN with 0.6% accuracy. In detail, Fig. 4 reveals that when there are fewer frequent terms, such as 5 or 8 or 10, the accuracy of r-Fretcit-kNN is comparable to that of Fretcit-kNN, but when there are more frequent terms, such as 12 or 15, the performance of Fretcit-kNN becomes apparently better than that of r-Fretcit-kNN. Exploring why the increase of the number of frequent terms benefits Fretcit-kNN much more than r-Fretcit-kNN might be helpful for developing better

algorithms, which is an important issue to study in the future.

6 Conclusion

This paper describes the first attempt of applying multi-instance learning techniques to web mining, which exhibits a new way to the solution of web mining tasks. In detail, the problem of web index recommendation is considered as a multi-instance problem by regarding index pages as bags while their linked pages as instances. This problem is then solved by extended kNN algorithms that employ the Minimal Hausdorff distance between ranked or unranked frequent term sets and utilize both the references and citers of an unseen bag in determining its label. More importantly, this paper shows that considering the multi-instance nature of such a problem is beneficial to the design of algorithms attaining good results.

In the presented work, only the whole web index pages could be recommended. That is, the proposed algorithms could only predict that whether a new web index page may interest the user or not. To develop some mechanism to locate the concrete linked pages that interested the user is an interesting issue for future work. Moreover, experiments reported in this paper show that the increase of the number of frequent terms used to describe the web pages benefits Fretcit-kNN much more than r-Fretcit-kNN. Exploring the reason for this phenomenon might be helpful for developing better algorithms, which is another interesting issue for future work. Furthermore, algorithms presented in this paper employ variants of Hausdorff distance to measure the distance between different objects. It may be possible to extend the algorithms through adopting other kinds of distance measures so that numerical attributes, ordered discrete attributes and unordered discrete attributes could be processed together, therefore the resulted algorithms could be applied to more tasks besides web index recommendation.

Acknowledgement

The comments and suggestions from the anonymous reviewers greatly improved this paper. This work was supported by the National Outstanding Youth Foundation of China under the Grant No. 60325207, the National Natural Science Foundation of China under the Grant No. 60105004, and the National 973 Fundamental Research Program of China under the Grant No. 2002CB312002.

References

[1] D.W. Aha. Lazy learning: special issue editorial. *Artificial Intelligence Review*, vol.11, no.1–5, pp.7–10, 1997.

[2] R.A. Amar, D.R. Dooly, S.A. Goldman, and Q. Zhang. Multiple-instance learning of real-valued data. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, pp.3–10, 2001.

[3] P. Auer. On learning from multi-instance examples: empirical evaluation of a theoretical approach. In *Proceedings of the 14th International Conference on Machine Learning*, Nashville, TN, pp.21–29, 1997.

[4] P. Auer, P.M. Long, and A. Srinivasan. Approximating hyper-rectangles: learning and pseudo-random sets. *Journal of Computer and System Sciences*, vol.57, no.3, pp.376–388, 1998.

[5] A. Blum and A. Kalai. A note on learning from multiple-instance examples. *Machine Learning*, vol.30, no.1, pp.23–29, 1998.

[6] Y. Chevaleyre and J.-D. Zucker. Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem. In E. Stroulia and S. Matwin, Eds. *Lecture Notes in Artificial Intelligence 2056*, Berlin: Springer, pp.204–214, 2001.

[7] B.V. Dasarathy. *Nearest Neighbor Norms: NN Pattern Classification Techniques*, Los Alamitos, CA: IEEE Computer Society Press, 1991.

[8] L. De Raedt. Attribute-value learning versus inductive logic programming: the missing links. In D. Page, Ed. *Lecture Notes in Artificial Intelligence 1446*, Berlin: Springer, pp.1–8, 1998

[9] T.G. Dietterich, A. Jain, R.H. Lathrop, and T. Lozano-Pérez. A comparison of dynamic reposing and tangent distance for drug activity prediction. In J. Cowan, G. Tesauro, and J. Alspector, Eds. *Advances in Neural Information Processing Systems 6*, San Mateo: Morgan Kaufmann, pp.216–223, 1994.

[10] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, vol.89, no.1–2, pp.31–71, 1997.

[11] G.A. Edgar. *Measure, Topology, and Fractal Geometry*, Berlin: Springer, 1990.

[12] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, Nashville, TN, 1997, pp.143–151.

[13] P.M. Long and L. Tan. PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, vol.30, no.1, pp.7–21, 1998.

[14] O. Maron. Learning from ambiguity. PhD dissertation, Department of Electrical Engineering and Computer Science, MIT, Jun. 1998.

[15] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In M.I. Jordan, M.J. Kearns, and S.A. Solla, Eds. *Advances in Neural Information Processing Systems 10*, Cambridge, MA: MIT Press, pp.570–576, 1998.

[16] O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, pp.341–349, 1998.

[17] S. Ray and D. Page. Multiple instance regression. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, pp.425–432, 2001.

[18] G. Ruffo. Learning single and multiple instance decision tree for computer security applications. PhD dissertation, Department of Computer Science, University of Turin, Torino, Italy, Feb. 2000.

[19] J. Wang and J.-D. Zucker. Solving the multiple-instance problem: a lazy learning approach. In *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, CA, pp.1119–1125, 2000.

[20] C. Yang and T. Lozano-Pérez. Image database retrieval with multiple-instance learning techniques. In *Proceedings of the 16th International Conference on Data Engineering*, San Diego, CA, pp.233–243, 2000.

[21] Q. Zhang and S.A. Goldman. EM-DD: an improved multi-instance learning technique. In T.G. Dietterich, S. Becker, and Z. Ghahramani, Eds. *Advances in Neural Information Processing Systems 14*, Cambridge, MA: MIT Press, pp.1073–1080, 2002.

[22] Q. Zhang, W. Yu, S.A. Goldman, and J.E. Fritts. Content-based image retrieval using multiple-instance learning. In *Proceedings of the 19th International Conference on Machine Learning*, Sydney, Australia, pp.682–689, 2002.

[23] Z.-H. Zhou and M.-L. Zhang. Neural networks for multi-instance learning. Technical Report, AI Lab, Computer Science & Technology Department, Nanjing University, Nanjing, China, Aug. 2002.

[24] Z.-H. Zhou and M.-L. Zhang. Ensembles of multi-instance learners. In N. Lavrac, D. Gamberger, H. Blockeel, and L. Todorovski, Eds. *Lecture Notes in Artificial Intelligence 2837*, Berlin: Springer, pp.492–502, 2003.

[25] J.-D. Zucker and J.-G. Ganascia. Changes of representation for efficient learning in structural domains. In *Proceedings of the 13th International Conference on Machine Learning*, Bary, Italy, pp.543–551, 1996.

[26] J.-D. Zucker and J.-G. Ganascia. Learning structurally indeterminate clauses. In D. Page, Ed. *Lecture Notes in Artificial Intelligence 1446*, Berlin: Springer, pp.235–244, 1998.

Appendix

Tables 2 to 6 present the detailed experimental results discussed in Section 5. In the tables 'Accu.' denotes accuracy, 'Preci.' denotes precision, and 'GM' denotes geometrical mean.

Table 2
Detailed experimental results when 5 frequent terms are used to describe the instances.

Data set	Txt-kNN			Cit-kNN			Fretcit-kNN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
V1	.895	.250	.500	.868	.250	.333	.921	.500	.667
V2	.737	.667	.182	.842	.667	.286	.921	.667	.500
V3	.632	.571	.267	.632	.286	.182	.868	.571	.667
V4	.868	.970	.889	.921	.939	.969	.895	.939	.939
V5	.711	1.00	.711	.789	.815	.880	.895	1.00	.871
V6	.816	1.00	.806	.895	.966	.903	.816	.897	.867
V7	.579	.688	.500	.605	.500	.533	.658	.625	.588
V8	.553	.800	.552	.579	.550	.611	.658	.650	.684
V9	.632	.556	.625	.632	.389	.700	.658	.556	.667
GM	.714	.722	.559	.751	.596	.600	.810	.712	.717

(Table 2 continued)

TFIDF			r-Txt-kNN			r-Cit-kNN			r-Fretcit-kNN		
Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
.816	.788	1.00	.895	.500	.500	.921	.500	.667	.921	.500	.667
.868	.852	.958	.842	.333	.200	.842	.667	.286	.868	.667	.333
.763	.793	.885	.789	.286	.400	.816	.143	.500	.868	.714	.625
.789	.000	.000	.868	1.00	.868	.842	.909	.909	.816	.879	.906
.895	.333	.333	.737	.889	.774	.632	.630	.810	.842	1.00	.818
.763	.429	.375	.763	1.00	.763	.816	.966	.824	.816	.931	.844
.842	.688	.917	.605	.750	.522	.632	.563	.563	.737	.688	.688
.474	.850	.500	.579	.650	.591	.605	.600	.632	.684	.600	.750
.605	.350	.778	.395	.667	.414	.553	.500	.529	.763	.556	.909
.757	.565	.638	.719	.675	.559	.740	.609	.636	.813	.726	.727

Table 3
Detailed experimental results when 8 frequent terms are used to describe the instances.

Data set	Txt-kNN			Cit-kNN			Fretcit-kNN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
V1	.868	.500	.400	.895	.500	.500	.895	.500	.500
V2	.816	1.00	.300	.763	.667	.200	.868	.667	.333
V3	.763	.429	.375	.816	.429	.500	.842	.571	.571
V4	.895	.970	.914	.789	.848	.903	.789	.788	.963
V5	.711	.963	.722	.789	.852	.852	.842	.963	.839
V6	.816	1.00	.806	.789	.897	.839	.842	.966	.848
V7	.605	.875	.519	.789	.813	.722	.763	.813	.684
V8	.579	.850	.567	.789	.800	.800	.632	.600	.667
V9	.553	.833	.517	.632	.333	.750	.605	.333	.667
GM	.734	.824	.569	.783	.682	.674	.786	.689	.675

(Table 3 continued)

TFIDF			r-Txt-kNN			r-Cit-kNN			r-Fretcit-kNN		
Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
.816	.788	1.00	.816	.500	.286	.816	.250	.200	.895	.500	.500
.868	.852	.958	.711	.333	.100	.842	1.00	.333	.789	.667	.222
.816	.793	.958	.711	.143	.167	.868	.429	.750	.868	.714	.625
.816	.500	.286	.895	.970	.914	.868	.939	.912	.842	.879	.935
.632	.333	.077	.763	.926	.781	.684	.704	.826	.816	.963	.813
.684	.429	.273	.816	1.00	.806	.842	.966	.848	.737	.897	.788
.684	.813	.591	.605	.750	.522	.632	.688	.550	.763	.750	.706
.605	.600	.632	.605	.700	.609	.684	.750	.682	.711	.655	.765
.632	.650	.650	.526	.722	.500	.658	.556	.667	.684	.444	.800
.728	.640	.603	.716	.672	.521	.766	.698	.641	.789	.719	.684

Table 4
Detailed experimental results when 10 frequent terms are used to describe the instances.

Data set	Txt-kNN			Cit-kNN			Fretcit-kNN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
V1	.842	.250	.250	.895	.250	.500	.895	.500	.500
V2	.816	1.00	.300	.684	.333	.091	.842	.667	.286
V3	.789	.429	.429	.842	.429	.600	.842	.571	.571
V4	.921	1.00	.917	.816	.848	.933	.842	.848	.966
V5	.737	.926	.758	.737	.852	.793	.842	.963	.839
V6	.789	1.00	.784	.789	.897	.839	.868	.966	.875
V7	.526	.563	.450	.658	.625	.588	.789	.750	.750
V8	.421	.550	.458	.632	.600	.667	.711	.655	.765
V9	.474	.611	.458	.605	.278	.714	.632	.444	.667
GM	.702	.703	.534	.740	.568	.636	.807	.707	.691

(Table 4 continued)

TFIDF			r-Txt-kNN			r-Cit-kNN			r-Fretcit-kNN		
Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
.816	.788	1.00	.895	.500	.500	.868	.500	.400	.895	.500	.500
.868	.852	.958	.816	1.00	.300	.868	.667	.333	.842	.667	.286
.816	.793	.958	.789	.286	.400	.842	.143	1.00	.868	.714	.625
.895	.500	.500	.895	.970	.914	.868	.909	.938	.816	.848	.933
.684	.333	.091	.737	.926	.758	.658	.667	.818	.842	1.00	.818
.711	.429	.300	.816	1.00	.806	.816	.931	.844	.789	.931	.818
.737	.813	.650	.632	.813	.542	.737	.688	.688	.816	.813	.765
.684	.750	.682	.553	.650	.565	.711	.650	.765	.711	.600	.800
.632	.700	.636	.605	.778	.560	.579	.333	.600	.684	.333	1.00
.760	.662	.642	.749	.769	.594	.772	.610	.710	.807	.712	.727

Table 5
Detailed experimental results when 12 frequent terms are used to describe the instances.

Data set	Txt-kNN			Cit-kNN			Fretcit-kNN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
V1	.816	.250	.200	.868	.250	.333	.921	.750	.600
V2	.816	1.00	.300	.789	.667	.222	.842	.667	.286
V3	.763	.571	.400	.789	.429	.429	.868	.714	.625
V4	.921	1.00	.917	.789	.788	.963	.816	.848	.933
V5	.789	1.00	.771	.658	.741	.769	.789	.889	.828
V6	.789	1.00	.784	.789	.931	.818	.895	.966	.903
V7	.526	.688	.458	.632	.625	.556	.763	.563	.818
V8	.579	.600	.600	.632	.450	.750	.711	.650	.765
V9	.553	.778	.519	.737	.500	.900	.711	.556	.769
GM	.728	.765	.550	.743	.598	.638	.813	.734	.725

(Table 5 continued)

TFIDF			r-Txt-kNN			r-Cit-kNN			r-Fretcit-kNN		
Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
.816	.788	1.00	.868	.250	.333	.816	.250	.200	.895	.500	.500
.842	.815	.957	.737	1.00	.231	.842	.667	.286	.842	.667	.286
.816	.793	.958	.763	.429	.375	.816	.286	.500	.868	.714	.625
.789	.500	.250	.895	.970	.914	.895	.939	.939	.842	.848	.966
.684	.333	.091	.737	.926	.758	.658	.667	.818	.737	.852	.793
.711	.571	.333	.816	1.00	.806	.789	.931	.818	.816	.966	.824
.842	.688	.917	.658	.875	.560	.684	.688	.611	.737	.688	.688
.737	.800	.727	.632	.750	.625	.711	.700	.737	.763	.700	.824
.579	.550	.611	.605	.667	.571	.711	.444	.889	.684	.389	.875
.757	.649	.649	.746	.763	.575	.769	.619	.644	.798	.703	.709

Table 6
Detailed experimental results when 15 frequent terms are used to describe the instances.

Data set	Txt-kNN			Cit-kNN			Fretcit-kNN		
	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
V1	.895	1.00	.500	.868	.250	.333	.921	.750	.600
V2	.789	1.00	.273	.763	.667	.200	.868	.667	.333
V3	.684	.857	.353	.737	.429	.333	.868	.571	.667
V4	.842	.939	.886	.763	.758	.962	.895	.909	.968
V5	.711	1.00	.711	.816	.889	.857	.895	.963	.897
V6	.763	1.00	.763	.816	.931	.844	.895	.966	.903
V7	.526	.750	.462	.737	.750	.667	.816	.750	.800
V8	.816	.850	.810	.789	.750	.833	.737	.650	.813
V9	.632	.833	.577	.658	.444	.727	.632	.444	.667
GM	.740	.914	.593	.772	.652	.640	.836	.741	.739

(Table 6 continued)

TFIDF			r-Txt-kNN			r-Cit-kNN			r-Fretcit-kNN		
Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.	Accu.	Recall	Preci.
.947	.939	1.00	.895	1.00	.500	.842	.250	.250	.895	.500	.500
.868	.815	1.00	.605	1.00	.167	.658	.667	.143	.842	.667	.286
.816	.862	.893	.789	.429	.429	.842	.429	.600	.895	.714	.714
.789	.500	.250	.921	1.00	.917	.842	.909	.909	.816	.848	.933
.684	.333	.091	.711	.926	.735	.711	.741	.833	.789	.926	.806
.711	.571	.333	.816	1.00	.806	.816	.966	.824	.842	.966	.848
.711	.313	1.00	.737	.875	.636	.711	.625	.667	.816	.813	.765
.658	.350	1.00	.632	.800	.615	.816	.800	.842	.711	.600	.800
.553	.900	.545	.632	.667	.600	.684	.333	1.00	.711	.444	.889
.749	.620	.679	.749	.855	.601	.769	.636	.674	.813	.720	.727