Web Clustering Based On Tag Set Similarity

Zhou Jingli
College of Computer Science & Technology
Huazhong University of Science & Technology, Wuhan, China
Email: jlzhou@mail.hust.edu.cn

Nie Xuejun
College of Computer Science & Technology
Huazhong University of Science & Technology, Wuhan, China
Email: niexuejun@sina.com

Qin Leihua*
College of Computer Science & Technology
Huazhong University of Science & Technology, Wuhan, China
Email: lhqin@mail.hust.edu.cn

Zhu Jianfeng
College of Computer Science & Technology
Huazhong University of Science & Technology, Wuhan, China
Email: jfzhou.hust@gmail.com

Abstract—**Tagging is a service that allows users to associate a set of freely determined tags with web content. Clustering web documents with tag sets can eliminate the time-consuming preprocess of word stemming. This paper proposes a novel method to compute the similarity between tag sets and use it as the distance measure to cluster web documents into groups. Major steps in this method include computing a tag similarity matrix with set-based vector space model, smoothing the similarity matrix to obtain a set of linearly independent vectors and compute the tag set similarity based on these vectors. The experimental results show that the proposed tag set similarity measures surpasses other common similarity measures not only in the reliable derivation of clustering results, but also in clustering accuracies and efficiencies.**

Index Terms—**web clustering, tag similarity matrix, similarity smoothing, tag set similarity**

I. INTRODUCTION

Web clustering has been an important tool for Web search engines [1-5]. With a good clustering method, the search results can be automatically organized into meaningful clusters, which enable efficient browsing and navigation. Numerous mathematical approaches have been made to detect similarities between web documents inside a cluster. Usually, these approaches use the vector space model to represent web documents. In this model, each web document is preprocessed into a bag of tags and is treated as a vector in a finite-dimensional space [6]. The cosine of the angle between two vectors is then used as a measure of their similarity. This vector space model for similarity computation has been extensively studied in the past, and a variety of methods have been proposed [7, 8]. However, the vector space model suffers the time-consuming preprocess, which is also referred to as word stemming or word segmentation. Additionally, this model can not be applied in the context of multimedia or other material that can not be processed in to bags of words.

With the success of tagging services like Flickr (http://www.fickr.com), del.icio.us (http://delicious.com) and YouTube (http://www.youtube.com), tags have become a powerful means to characterize the content of web documents. Tags are collections of keywords attached by users to help describe the web document, which is also referred to as tag sets. Clustering web documents through tag sets is a promising way, because it eliminates the need for time-consuming word stemming or word segmentation. However, the similarity measures in traditional web clustering model are not directly applicable to the web clustering problem, because tag sets usually consist of relatively small number of tags compared to the number of terms in webs considered in information retrieval literature. To deal with this dilemma, Bruno Pôssas etc. [9] proposed a set-based vector model to compute the correlations among index terms in a web document, which can effectively improve the retrieval

* To whom correspondence should be addressed.

doi:10.4304/jcp.6.1.59-66

effectiveness for general web collections. Although its performance improvements over the traditional vector space model are significant, it requires computation of exponential complexity, and therefore is not computationally feasible for the web clustering environment where the web search collections are usually so large and fluid. Other approaches suffer the similar problems [10-19].

In this paper, we propose a novel web clustering method to compute the similarities between tag sets based on tag similarity information. Firstly, we gather all tags into a tag vocabulary and compute a tag matrix with a simplified set-base vector space model. Each element in the matrix represents a similarity between two tags. Next, a similarity smoothing is performed to obtain a set of linearly independent vectors, which incorporate the implicit relationships among tags that are ignored in the set-based vector space model. Finally, the tag set similarity is computed based on these vectors. We conduct experiment on web corporal crawled from two web sites with tagging service enabled, and compare the performance of tag set similarity measure against other common similarity measures. As the experimental results show, the tag set similarity surpasses those similarity measures not only in the reliable derivation of web clustering results, but also in web clustering accuracies and efficiencies.

The remaining of this paper is structured as follows. Section Ⅱ describes the definitions and calculations of tag set similarity. In section Ⅲ, we describe our evaluation process and present the experimental results using two web corporal. The paper is concluded in section Ⅳ.

II. TAG SET SIMILARITY

In this section, we will detail the steps for computing tag set similarity. Firstly, we execute a supervised learning process to compute the tag similarity matrix with a simplified set-based vector space model. Secondly, we perform a smoothing process on the similarity matrix and obtain a set of linearly independent vectors. Finally, the tag set similarity is computed based on these vectors.

Without loss of generality, we assume that we have a collection of training web documents $W = \{w_1, w_2, ..., w_n\}$. For each web document w_i, there is a tag set $s_i = \{t_{i0}, t_{i1}, ..., t_{il}\}$ associated with it, where t_{ik} ($0 \le k \le l$) is a tag included in s_i. The tag vocabulary V is the set of all tags associated with web documents in W. V can be simply computed by performing a union operation on all tag sets, $V = s_0 \cup s_1 \cup ... \cup s_n = \{t_0, t_1, ..., t_m\}$.

A. Tag Similarity Matrix

To begin with, we should compute the similarity between each pair of tags within the tag vocabulary V. In traditional vector space models, it is assumed that the terms are mutually independent. This assumption is often made for mathematical convenience and simplicity of implementation. However, it is generally accepted that exploitation of the correlation among terms might be used to improve retrieval effectiveness with general collections. One of such models is the set-based vector space model, which takes into account patterns of term co-occurrence and is efficient enough to be of practical value. Its components are term sets, where a term set is a set of the index terms of a collection of webs. The model exploits the intuition that semantically related terms often occur close to each other by implementing a pruning strategy that restricts computation to proximate term sets. Although this model can achieve significant performance improvements over the traditional vector space model, it require computation of exponential complexity, and therefore are not computationally feasible for the considered web clustering environment where the number of tags is huge and tag clouds are dynamically updated.

In our case, we treat each tag as a term and each tag set as a term set, and proposed a simplified set-based vector space model to apply in web clustering environment. Firstly, we introduce several key definitions as follows:

Definition 1 Occurrence Frequency of a Tag

Given a tag t_i, the occurrence frequency of the tag is the number of tag sets that contain t_i, denoted as f_i.

Definition 2 Tag Pair

A tag pair is a pair of different tags t_i and t_j, denoted as $t_{i,j} = \{t_i, t_j\}$, where $i = 1,2,...,n$, $j = 1,2,...,n$ and $i < j$.

Apparently, there are $\frac{n \times (n-1)}{2}$ unique tag pairs for the tag vocabulary V.

Definition 3 Occurrence Frequency of a Tag Pair

Given a tag pair $t_{i,j}$, the occurrence frequency of the tag pair is the number of tag sets that contain both tags t_i and t_j, denoted as $f_{i,j}$. $f_{i,j} = 0$ means that t_i and t_j do not simultaneously appear in any tag set.

With above definitions, we defined the tag similarity between two tags t_i and t_j as follows:

$$c_{i,j} = \frac{f_{i,j}}{f_i + f_j - f_{i,j}} \qquad (1)$$

Where $c_{i,j}$ is the tag similarity between tags t_i and t_j, f_i and f_j are the occurrence frequency of these two tags respectively, $f_{i,j}$ is the occurrence frequency of the tag pair $t_{i,j}$.

For each tag pair, there exists a tag similarity. If we treat the index of first component in the tag pair as the row index, and the index of second component in the tag pair as the column index, all tag similarities can form a matrix, which we refer to as tag similarity matrix,

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{pmatrix}$$

The matrix can be denoted as $C = (c_{i,j})_{n\times n}$, where $0 \le c_{i,j} \le 1$ and $c_{i,i} = 1$, $i = 1,2,...,n$, $j = 1,2,...,n$. It is clear that $c_{i,j} = c_{j,i}$, which means that C is a symmetric matrix.

B. Similarity Smoothing

The similarity represented by $c_{i,j}$ is also referred to as *explicit similarity*, because it is computed based on the co-occurrence of two tags. However, there exists another similarity, which is referred to as *implicit similarity*. Consider two tag set $s_0 = \{t_0, t_1, t_2\}$ and $s_1 = \{t_0, t_1, t_3\}$, according to formula (1), the similarity between every two tags are computed as follows:

$$c_{0,1} = c_{1,0} = 1$$

$$c_{0,2} = c_{2,0} = 0.5$$

$$c_{0,3} = c_{3,0} = 0.5$$

$$c_{1,2} = c_{2,1} = 0.5$$

$$c_{1,3} = c_{3,1} = 0.5$$

$$c_{2,3} = c_{3,2} = 0$$

From these similarities, we may conclude that there is none relationship between tags t_2 and t_3. However, this is not the case. Although t_2 and t_3 do not appear simultaneously in any tag set, t_2 and $\{t_0, t_1\}$ appear in the same tag set, thus there should be some relationship between t_2 and $\{t_0, t_1\}$. Similarly, there also should be some relationship between t_3 and $\{t_0, t_1\}$. By transitivity, there should be some relationship between t_2 and t_3. However, the $c_{2,3}$ can not reflect this relationship. The root cause of this problem is because there is no globally agreed list of tags user can choose from, different users use different tags to describe similar web documents, and even a single user's tagging practice may vary over time [16]. Consequently, the implicit similarity is lost in the tag similarity matrix.

Therefore, we should find a method to compute the implicit similarity hidden in the similarity matrix. We referred this method as similarity smoothing, which is motivated by the similarity ranking algorithm proposed in [5].

The starting point of our approach is to treat each similarity value $c_{i,j}$ between two tags, t_i and t_j, as cosine of the angle $\theta_{i,j}$ between two vectors, v_i and v_j, such that $0 \le \theta_{i,j} \le \pi/2$, $\forall i, j$. That is, when two tags are identical, the angle will be zero, producing a maximum similarity measure. With this interpretation, our problem becomes to find a set of linearly independent vectors, $\{v_1, v_2, ..., v_n\}$, that satisfy the constraints $(v_i) \cdot (v_j) = \cos\theta_{i,j} = c_{i,j}$, $\forall i, j$. It has been proven in [5] that each vector will incorporate both explicit similarity and implicit similarity.

The set of linearly independent vectors can be obtained by applying orthogonal triangulation to the tag similarity matrix C. That is, we should compute a $n \times n$ matrix $T = ((v_1)^T, (v_2)^T, ..., (v_n)^T)$ such that $T^T T = C$. Based on whether C is a positive definite matrix or not, we will apply a standard or modified Cholesky decomposition algorithm respectively. Without loss of generality, we assume that all characteristic values of C are denoted as $\{\lambda_0, \lambda_1, ..., \lambda_n\}$.

Case 1: C is a positive definite matrix

In this case, all characteristic values of C are positive values, $\lambda_i > 0$, $0 \le i < n$. Therefore, there is an $n \times n$ diagonal matrix $D = diag(\lambda_i)$, and an $n \times n$ matrix L with independent columns such that $C = L^T DL$. Let $\sqrt{D}$ be

$$\sqrt{D} = \begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda_n} \end{pmatrix}$$

Since $(\sqrt{D}L)^T(\sqrt{D}L) = L^T \Lambda DL = C$, we can know that $T = \sqrt{D}L$. The matrix T can be obtained through a standard Cholesky decomposition algorithm, as shown in Fig. 1 with C-like pseudo code.

```
1 : i = 1 step 1 until n do
2 : j = 1 step 1 until n do
3 :         begin x = c_{i,j}
4 :             for k = i − 1 step −1 until 1 do
5 :                 x = x − c_{j,k} × c_{i,k}
6 :             if i == j then
7 :                 p_i = 1/√x
8 :             else
9 :                 c_{i,j} = x × p_i
10: end i, j
```

Figure 1. The standard Cholesky decomposition algorithm

Case 2: C is NOT a positive definite matrix

If C is not a positive definite matrix, there may be negative characteristic values, $\lambda_i \le 0$. Therefore, in the line 5 of the algorithm in Fig. 1, $x \le 0$, and in line 7 p_i will be a complex number, which is meaningless for tag similarity. To deal with this problem, we propose a modified Cholesky decomposition algorithm, as shown in Fig. 2.

```
1 : i = 1 step 1 until n do
2 : j = 1 step 1 until n do
3 : begin x = c_{i,j}
4 :   for k = i − 1 step −1 until 1 do
5 :     x = x − c_{j,k} × c_{i,k}
6 :   if i == j then
7 :     if x ≤ 0 then
8 :       set all elements in row i
            and column i to 0
9:        i = i + 1
10:       goto line 2
11:     else
12:       p_i = 1/√x
13:   else
14:     c_{i,j} = x × p_i
15: end i, j
```

Figure 2. The modified Cholesky decomposition algorithm

Compared with the algorithm in Fig. 1, we add a conditional statement (highlighted in bold font) to check whether x is less than or equal to 0, if the evaluation of the conditional statement is TRUE, we set all elements in row i and column i to 0 and continue the loop. The result matrix resembles a lower triangular matrix, except there are some rows where elements are all 0.

Each row in the result matrix T is referred to as a similarity vector of corresponding tag, where the row index is just equal to the index of the tag in tag vocabulary V. The vector has incorporated both explicit similarities and implicit similarities [5].

C. Tag Set Similarity

After perform smoothing on the tag similarity matrix, we obtain a set of linearly independent vectors and a set of zero vectors. Each tag is associated with a vector through the indexes of the tag and the vector. To compute the tag set similarity, we first compute the vector for the tag set. The most intuitive way to do this is to add up the vectors of all tags in a tag set. For example, given a tag set $s_0 = \{t_0, t_1, t_2\}$, we first find the corresponding vectors for each tag in the tag set , denoted as $\{v_0, v_1, v_2\}$, the vector for s_0 can be simply computed as follows:

$$v_{s_0} = v_0 + v_1 + v_2 \quad (2)$$

However, a practical issue arising when applying formula (2) is to deal with the tags whose corresponding vectors are {0} as a result of execution of the modified Cholesky algorithm in Fig. 2. While there are many possible variations of methods for this problem, we use a heuristic method in which the 0-vector is replaced with the vector for the tag that has the highest similarity to the 0-vector tag.

In the previous example, let's assume that $v_1 = \{0\}$ $c_{0,1} = 0.6$, $c_{1,2} = 0.3$, then we replace v_1 with v_0 and the vector for s_0 will be changed to $\{v_0, v_0, v_2\}$, thus $v_{s_0} = v_0 + v_0 + v_2$.

After the vectors for tag sets are obtained, we can choose similarity measures. The most commonly used similarity measures in web clustering include Euclidean distance and Cosine measure.

With Euclidean distance, the tag set similarity is defined as follows:

$$sim_E(s_i, s_j) = \sqrt{(\sum_{k=1}^{m} | x_{ik} - x_{jk} |^2)} \quad (2)$$

Where x_i is a component of v_{s_i}. The Euclidean distance is the standard metrics for geometrical problems.

Similarity can also be defined by the angle or cosine of the angle between two vectors. With Cosine measure, the tag set similarity is defined as follows:

$$sim_C(s_i, s_j) = \sum_{k=1}^{m} x_{ik} x_{jk} \Big/ \sqrt{\sum_{i=1}^{m} (x_{ik})^2 \sum_{i=1}^{m} (x_{jk})^2}$$

The Cosine measure can capture a scale invariant understanding of similarity.

III. EVALUATIONS

In this section, we describe the web corporal used for the performance evaluations, and compare the results against the performance achieved by traditional similarity measure based on bag-of-words method. The computer used for conducting experiments is equipped with a AMD dual-core processor at 2.21GHz, 1GB RAM, and 160GB disk. We choose the standard *k*-means clustering algorithm for clustering web documents.

A. Web Corporal

For the experiments, we used two different web collections to evaluate the effectiveness of tag set similarity approach in web clustering problem.

The first corpus is the web documents we have crawled from http://delicious.com. We treat each popular tag as a category, and 10 categories were selected: design, blog, video, software, tools, music, programming, webdesign, reference and tutorial. For each category, we have crawled the most recent 500 web pages in the date of 2010-1-1. Thus, every category contains 500 articles. For each web page, we remove the corresponding popular tag from its tag set. The total 5,000 web documents are associated with 54,758 tags, and 786 distinct tags. The average number of distinct tags per web document was 0.1136 with a minimum of 1 and a maximum of 12. To compare with traditional bag-of-words method, we perform the preprocess on the corpus, which includes ignoring all the words that contained digits or non alpha-numeric characters, removing words from a stop-word list containing 293 words, and filtering out low frequency words which appeared only once in entire set.

The second corpus is the web documents we have crawled from http://blog.sina.com.cn, and also 10 categories were selected: entertainment, sports, culture, finance, automobile, house, education, game, military and travel. For each category, we have crawled top 50 blog

files of each day from the date of 2010-1-1 to the date of 2010-1-10. The total 5,000 webs are associated with 32,614 tags, and 524 distinct tags. The average number of distinct tags per web document was 0.0681 with a minimum of 1 and a maximum of 9.The preprocessing included word segmentation, ignoring all the words that contained digits or non alpha-numeric characters, removing words from a stop-word list containing 717 words, and also filtering out low frequency words which appeared only once in entire set.

B. Evaluation Metrics

The testing web documents used for evaluating the proposed tag set method are formed by mixing webs from multiple clusters randomly selected from the web collections. At each run of the test, webs from a selected number k of topics are mixed, and the mixed web set, along with the cluster number k, are provided to the clustering process. Two metrics, the accuracy P and the mutual information metric MI, are used to measure the web clustering performance. For evaluating a single cluster, we use accuracy while the entire clustering is evaluated using mutual information.

Given L categories, $A=\{a_1,a_2,...,a_l\}$, and H clusters, $U=\{u_1,u_2,...,u_h\}$, let n_g be the number of webs in cluster u_g and n_g^k denote the number of webs in that cluster are classified to category a_k. The accuracy of cluster n_h can be defined as

$$P(C_l)=\frac{1}{n_g}\max(n_g^k)$$

While accuracy is suitable for measuring a single cluster's quality, it is biased to favor smaller clusters. Consequently, for the overall performance evaluation, we use a measure based on mutual information:

$$MI=\frac{1}{n}\sum_{g=1}^{h}\sum_{k=1}^{l}n_h^k\frac{\log(\frac{n_g^k n}{\sum_{i=1}^{h}n_i^k\sum_{i=1}^{l}n_g^i})}{\log(h\times l)}$$

Mutual information is a symmetric measure for the degree of dependency between the clustering and the categorization. We use the symmetric mutual information criterion because it successfully captures how related the clusters and categories are without a bias towards smaller clusters.

In addition, we also measure the time that both traditional bag-of-words method and tag set method have spent on the clustering process.

C. Result

Table Ⅰ and Table Ⅱ show the accuracy results with Euclidean measure and Cosine measure respectively. Table Ⅲ and Table Ⅳ show the mutual information results with Euclidean measure and Cosine measure respectively. Table Ⅴ and Table Ⅵ show the time spent on web clustering with Euclidean measure and Cosine measure respectively. The evaluations were conducted for the cluster numbers ranging from 2 to 10. For each given cluster number cn, 50 test runs were conducted on different randomly chosen clusters, and the final performance scores were obtained by averaging the scores from the 50 tests.

TABLE I. ACCURACY OBATINED WITH EUCLIDEAN MEASURE

cn	Delicious		Sina	
	Bag-of-Words	Tag Set	Bag-of-Words	Tag Set
2	0.832	0.852	0.785	0.813
3	0.742	0.853	0.699	0.753
4	0.685	0.775	0.639	0.705
5	0.697	0.734	0.654	0.712
6	0.643	0.727	0.621	0.686
7	0.628	0.684	0.585	0.654
8	0.619	0.647	0.584	0.634
9	0.658	0.683	0.602	0.675
10	0.674	0.717	0.611	0.652

TABLE II. ACCURACY OBATINED WITH COSINE MEASURE

cn	Delicious		Sina	
	Bag-of-Words	Tag Set	Bag-of-Words	Tag Set
2	0.987	0.995	0.932	0.942
3	0.953	0.986	0.852	0.898
4	0.932	0.954	0.832	0.843
5	0.853	0.934	0.754	0.815
6	0.792	0.875	0.731	0.763
7	0.775	0.853	0.697	0.743
8	0.714	0.832	0.651	0.684
9	0.743	0.842	0.642	0.695
10	0.754	0.831	0.632	0.675

TABLE III. MUTUAL INFORMATION OBATINED WITH EUCLIDEAN MEASURE

cn	Delicious		Sina	
	Bag-of-Words	Tag Set	Bag-of-Words	Tag Set
2	0.652	0.676	0.614	0.643
3	0.668	0.665	0.632	0.632
4	0.587	0.647	0.598	0.597
5	0.565	0.632	0.543	0.576
6	0.542	0.623	0.543	0.642
7	0.577	0.601	0.545	0.600
8	0.464	0.598	0.498	0.543
9	0.495	0.562	0.467	0.553
10	0.387	0.494	0.432	0.424

TABLE IV. MUTUAL INFORMATION OBATINED WITH COSINE MEASURE

cn	Delicious		Sina	
	Bag-of-Words	Tag Set	Bag-of-Words	Tag Set
2	0.931	0.953	0.825	0.843
3	0.904	0.923	0.742	0.796
4	0.854	0.899	0.732	0.754
5	0.782	0.874	0.668	0.743
6	0.733	0.842	0.665	0.681
7	0.721	0.823	0.642	0.684
8	0.694	0.775	0.612	0.631
9	0.712	0.784	0.673	0.663
10	0.732	0.801	0.623	0.623

TABLE V. TIME SPENT DURING CLUSTERING WITH EUCLIDEAN MEASURE (S)

cn	Delicious		Sina	
	Bag-of-Words	Tag Set	Bag-of-Words	Tag Set
2	198.55	143.15	201.52	160.04
3	247.79	196.64	249.85	207.60
4	301.53	248.74	299.85	258.06
5	346.95	310.04	350.85	302.34
6	398.59	348.52	402.54	349.67
7	453.35	403.85	445.95	404.08
8	511.43	444.53	505.45	457.54
9	549.96	502.85	561.62	503.98
10	607.34	548.52	591.76	561.34

TABLE VI. TIME SPENT DURING CLUSTERING WITH COSINE MEASURE (S)

cn	Delicious		Sina	
	Bag-of-Words	Tag Set	Bag-of-Words	Tag Set
2	229.43	167.53	227.13	160.35
3	274.52	223.42	274.54	219.25
4	329.85	278.94	332.86	272.26
5	384.32	340.05	391.96	334.35
6	434.65	387.88	445.52	381.09
7	496.75	439.85	491.54	446.84
8	552.85	495.84	553.53	494.83
9	605.42	553.52	608.52	548.54
10	657.98	603.86	667.09	601.85

Based on the results in these tables and figures, we can draw some conclusions:

(1) As far as accuracy and mutual information of clustering results are concerned, the similarity measure based on tag set can achieve better performance than the similarity measure based on bag-of-words, this is because the users who attach tags to web content always well understand the content and thus choose more appropriate tag to characterize them.

(2) The time spent on word stemming accounts for almost 20% of total time spent on web clustering when cluster number is 2 and the percentage decreases as the cluster number grow. Clustering web with similarity measure based on tag set eliminate the time-consuming word stemming, which can save much time.

(3) The improvement becomes more obvious for the Delicious corpus than the Sina corpus. This is because web clusters and tag sets in Delicious are generally more compact and focused than the clusters in Sina. The above experimental results for the two web corpora are mostly in line with the expectations because web clustering methods generally produce better results for web corpora comprised of compact and well-focused clusters and tag sets.

Next, we set the size of training set as 50, 100, 200, 400, and 800 respectively to cluster the two web corporal. The number of clusters cn was set to 10 and each setting was run 50 times o capture the random variation in results. The results are shown from Fig. 3 to Fig. 6.

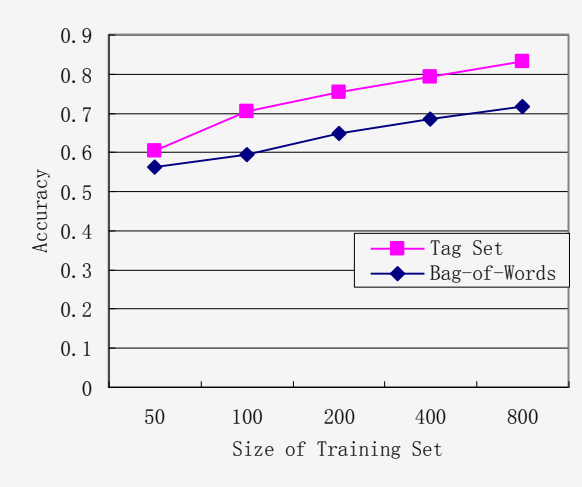

Figure 3. Accuracy under different sizes of training set with Euclidean measure

Figure 4. Accuracy under different sizes of training set with Cosine measure

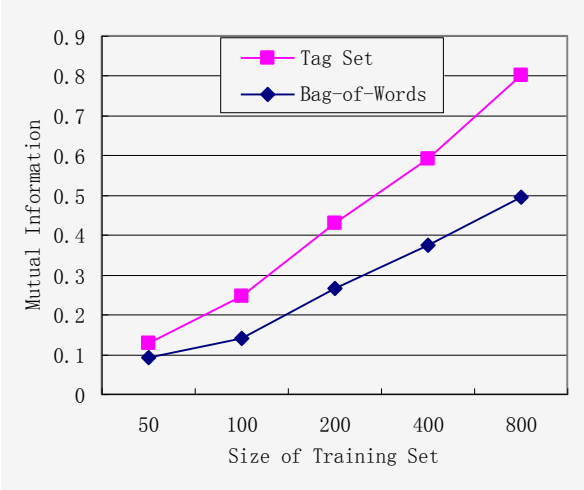

Figure 5. Mutual Information under different sizes of training set with Euclidean measure

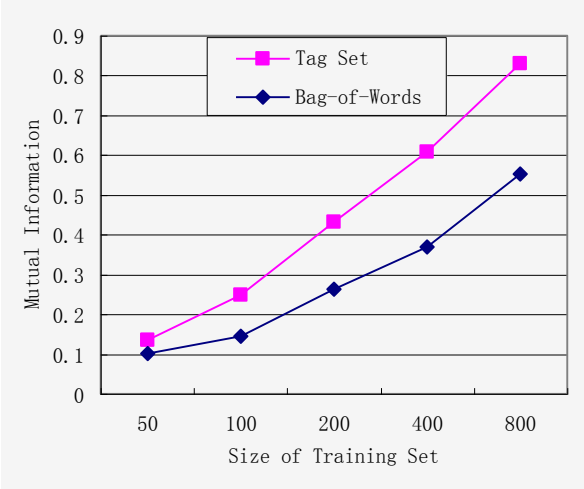

Figure 6. Mutual Information under different sizes of training set with Cosine measure

As we can see from these Figures, both accuracy and mutual information increase as the size of training set grows. Regardless of the size of training set and similarity measures, the performance of web clustering achieved based on tag set are better than the performance achieved based on bag-of-words.

IV. Conclusion

In this paper, we have presented a tag set similarity approach to address web clustering task. The tag set similarity effectively incorporates the tag similarity information, which can efficiently deal web documents with incomplete, vague or imprecise information. Besides that, it can eliminate the time-consuming preprocess of word stemming or word segmentation. We tested our method using both Delicious and Sina corpus. The experimental results show that performance of the proposed tag set similarity measurement surpasses traditional similarity measurement not only in the reliable derivation of clustering results, but also in clustering accuracies and efficiencies.

Acknowledgment

The work was supported by Ministry Foundation (“Research on Intelligent Storage System Based on Customizable Services”) and National Natural Science Foundation of China (No.60673001).

References

[1] C. Carpineto, S. Osiski, G. Romano and D. Weiss. “A Survey of Web Clustering Engines”. *ACM Computing Surveys*, v 41, n 3, p 17:1-17:38, July 2009.

[2] E. Fersini, E. Messina and F. Archetti. “A Probabilistic Relational Approach for Web document Clustering”. *Information Processing and Management*, v 46, n 2, p 117-130, March 2010.

[3] S. Nirkhi and K. N. Hande. “A Survey on Clustering Algorithm for Web Applications”. In: *Proceedings of the 2008 International Conference on Semantic Web and Web Services (SWWS 2008)*, p 124-129, July 2008.

[4] C. Carpineto , S. Mizzaro, G. Romano and M. Snidero. “Mobile information retrieval with search results clustering: Prototypes and evaluations”. *Journal of the American Society for Information Science and Technology*, v 60, n 5, p 877-95, May 2009.

[5] P. Jonghun, C. Byung-Cheon and K. Kwanho. “A Vector Space Approach to Tag Cloud Similarity Ranking. *Information Processing Letters*, v 110, n 7, p 1-8, March, 2010

[6] P. Ganesan, H. Garcia-Molina, J. Widom, “Exploiting hierarchical domain structure to compute similarity”. *ACM Transaction on Information Systems*, v 21, n 1, p 64-93, 2003.

[7] M.W. Berry, Z. Drmac, E.R. Jessup. “Matrices, vector spaces, and information retrieval”. *SIAM Review*, v 41, n 2, p 335-62, 1999.

[8] S.K.M. Wong, W. Ziarko, V.V. Raghavan, P.C.N. Wong. “On modeling of information retrieval concepts in vector spaces”. *ACM Trans. Database System*, v 12, n 2, p 299 - 321, 1987.

[9] P. Bruno, Z. Nivio, Meira Jr. Wagner and Ribeiro-Neto, Berthier. “Set-based vector model: An efficient approach for correlation-based ranking”. *ACM Transactions on Information Systems*, v 23, n 4, p 397-429, 2005.

[10] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, NewYork, March 1990.
[11] K. Kummamuru, A. Dhawale and R. Krishnapuram. "Fuzzy Co-Clustering of Webs and Keywords". In: *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, vol.2, p 772-7, 2003.
[12] O. Alonso and M. Gertz. "Clustering of search results using temporal attributes". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p 597–598, 2006
[13] C. Carpineto, S. Mizzaro, G. Romano and M. Snidero. "Mobile information retrieval with search results clustering: Prototypes and evaluations". *Journal of the American Society for Information Science and Technology*, v 60, n 5, p 877-95, May 2009.
[14] H. Chen and S. Dumais. "Bringing order to the Web: Automatically categorizing search results". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p 145–152, 2000.
[15] X. Liu and B. W. Croft. "Representing clusters for retrieval". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p 671–672, 2006.
[16] G. Begelman, P. Keller and F. Smadja. "Automated tag clustering: Improving search and exploration in the tag space". In : *Proceedings of Collaborative Web Tagging Workshop at WWW* , May 23-26, 2006.
[17] S. Osinski, J. Stefanowski, D.Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition". In: *Proceedings of the International Conference on Intelligent Information Systems (IIPWM)*, 2004.
[18] B. Stein, S. Meyer zu EISSEN and F. Wibrock. "On cluster validity and the information need of users". In *Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA)*, 216–221, 2003.
[19] J. Stefanowski and D. Weiss. "Carrot^2 and language properties in Web search results clustering". In: *First International Atlantic Web Intelligence Conference and Lecture Notes in Artificial Intelligence* Vol.2663, p 240-9, 2003.
[20] Y.C. Zhang and G.D. Xu. "Using Web clustering for Web communities mining and analysis". In: *ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops*, p 20-31, 2008.
[21] C.M. Lu, C. Xin, and E.K. Park. "Exploit the tripartite network of social tagging for web clustering". In: *Proceeding of International Conference on Information and Knowledge Management*, p 1545-1548, 2009.

Zhou Jingli, born in 1946. She received the B.E. degree in 1969. She is a Professor and doctor advisor at Huazhong University of Science and Technology. She had been a visiting scholar in USA from 1995 to 1996 and has been honor of the State Department Special Allowance since 1999. Her main research interests include high performance storage technology and system, multimedia computing and communication, network security.

Nie Xuejun, born in 1979. He received the M.S degree in 2002. He is a Ph.D. Candidate at Huazhong University of Science and Technology. His main research interests include content aware storage system and network storage.

Qin Leihua, born in 1968. He received the Ph.D degree in 2008. He is a assistant professor at and associate professor Huazhong University of Science and Technology. His main research interests include network security and network storage.

Zhu Jianfeng, born in 1978. He received M.S degree in 2005. He is a Ph.D. Candidate at Huazhong University of Science and Technology. His main research interests include network storage and disaster tolerance.