Classifying Blog Posts with Tag Propagation

Bingquan Liu[1], Baoxun Wang[1], Zhen Xu[1], Xiaolong Wang[1], Peng Li[2]
[1]Intelligent Technology & Natural Language Processing Lab, Harbin Institute of Technology, Harbin 150001, China
[2]Harbin University of Science and Technology, Harbin 150001, China
Email: {liubq, bxwang, zxu, xlwang} @insun.hit.edu.cn, pli@hrbust.edu.cn

Abstract

Blog tags are usually considered to be supplementary information for blog post classification tasks. Due to the sparsity of tag features, improving performance of classifiers merely using tags is not a trivial operation. This paper presents a blog post classification approach based on the tag propagation strategy. Using a dataset of blog posts gleaned from the Internet, tags of a blog post are propagated from tags of its K nearest neighbors in the blog post dataset. In this case, the original binary feature vectors are changed to real-value ones and the sparsity is reduced. Experimental results show that the classification method based on the tag propagation strategy obtains good performance.

Key words

Blog post classification; Tag feature; Tag propagation; Sparsity reduction

1. Introduction

Blogs provide platforms for web users to post personal diaries arranged in reverse chronological order and updated them frequently with new information on particular topics. The value of blogs as a source of web information has been recognized in recent years. Many studies have been devoted to handling the blogs in different areas such as text mining, information retrieval, and social networks.

As a meaningful way to manage and mine information in blog posts, blog post classification has attracted much attention of researchers. The task aims to assign one blog post to a pre-defined category (Sun et al. 2007). Blog post classification is usually based on techniques of text classification, but there are some particular aspects due to the particular structure of blog post. Firstly, blog posts usually contain tags marked by the author, which provide a degree of semantic information. Secondly, blog posts usually include hyperlinks implying some recommended contents and correlation between contents, and provide many heuristic features to classify blog posts. Finally, blog posts are HTML/XHTML text, and contain a substantial number of HTML tags. These characteristics indicate that the blog post classification task is different in comparison with common text classification.

Blog post tags are typical user-generated information with great potential for the blog post classification task (Tsai and F.S 2011). Delivered by the users themselves, the tags usually indicate the author's descriptions with regard to a given post. Thus it is possible for tags to be taken as the important features for the classifiers, which also makes this task different from traditional text classification. This paper attempts to build classifiers using the tags as features, so as to investigate the effect of the blog tags. However, it is common

to observe a blog post holding a small number of tags, thus it is possible for classification models based on the ordinary tag features to suffer from the problem of feature sparseness (Sood et al. 2007). To improve the performance of a blog post classifier, it is necessary to adopt complementary information to reduce the feature sparseness, which is the major motivation of this paper. Further, there are some detailed problems to solve, such as accurate tag extraction, synonymous tag reorganization, etc.

In this paper, we propose a tag-propagation-based method to classify blog posts. According to content similarity of blog posts, tags are propagated to reduce the sparseness of the tag feature space. By introducing real-value feature vectors based on the propagation strategy, the performance of the classifiers (trained with support vector machine) has been improved. In addition, this paper also discusses the technique for adaptive tag extraction.

The rest of the paper is organized as follows. Section 2 is a survey of related works; our tag propagation algorithm is presented in Section 3; the experimental results are shown in Section 4 and the conclusions are recorded in Section 5.

2. Related Work

Tagging allows systems to collocate relevant information, and also provides users a way to find related information (Mathes 2004). It allows users to assign keywords (tags) to annotate links as useful resources, and facilitate their future access by a tag creator (Macgregor et al. 2006). Tagging systems are currently becoming more and more popular. These systems enable users to add keywords (i.e., “tags”) to Internet resources without reference to any controlled vocabulary. Tagging systems introduce new ways for communication and opportunities for data mining, and have the potential ability to improve searches, spam detection, reputation systems, etc. Marlow et al. (2006) offers the taxonomy of collaborative tagging systems.

Berendt et al. (2007) compared the performance of blog post classification using features derived from tags, a title, and a body. They used the WordNet domain (WND) label system in multi-annotator classification on a blog corpus and developed a system of text-classification methods. Based on this, they offer empirical evidence to argue that tags are not metadata, but are “more content”. They proved that tags have a low similarity to a post body, and that tags together with the body yield better classification accuracy than either of them alone. Brooks et al. (2006) studied the issue of tags’ ability to describe semantics of a blog post and also the effectiveness of using tags to categorize blog posts. They found that tags are useful in grouping blog posts into broad categories, but are less effective in representing the content of blog posts. The authors then extracted the top three words with the highest TF*IDF scores from each blog post to represent it, and developed better categorization. In the study of Hayes et al. (2007), it is observed that frequently occurring tags are usually good meta-labels of a cluster produced using content clustering.

A text feature building strategy for image classification has been presented by Wang et al. (2009), which is very similar to our idea. In their work, the visual features and the associated texts of the images are extracted. Given an image, the visual features are extracted to find the K most similar images, of which associated texts are adopted to build text features for the given image. The content oriented feature propagation strategy has shown significant potential in both the image and the text classification problems.

3. Approach

The traditional text classification approaches primarily focus on the information provided by the text bodies. For the task of blog post classification, user-generated tags are of great value in labeling and summarizing the contents of the posts. This paper proposes a method of exploring the power of user-generated tags in blog classification task. Nevertheless, there are some problems that influence the performance of the classifier, as follows. (1) Extracting tags from blog posts is a nontrivial work, because tags are embedded in blog posts and different web sites have different structures. Using tags as features of blog classification, tag extraction is an important strategy in our approach. In Section3.1, this paper will introduce the method used in our experiment to extract tags from blog posts.(2) Blog posts generally have few tags, usually less than 5 and this paper will give the distribution of tag numbers in Section 3.2. Because of the above problem, the tag features are very sparse with binary items (items with a value of 0 or 1). Obviously, the sparse binary features cannot offer sufficient information of the classifiers so the performance of the post classification is not satisfactory. Thus, in Section 3.3, this paper proposes a method to complement the tag features based on content similarity of posts and turn the binary features into real-value ones, so as to improve the precision of the blog post classification methodologies using only tags.

3.1 Tag Extraction

The purpose of this paper is to propagate tag features comprising tags of blog posts and tags that cannot be directly obtained from blog posts. Tag extraction plays an important role in this paper. Tag extraction involves determining a starting position and an ending position of a tag region that contains tags of post. There are several methods to extract tag information from web pages. A template-based method is one of simplest and most effective methods used to extract information from web pages (Vargas-Vera 2001; Collier 1996). Since we focus on tag propagation and a tag region is always commented by html tags such as “ ”, therefore we adopt template-based method in tag extraction process. After analyzing the structures of 10 blog sites, we designed several templates that can extract tags from blog posts.

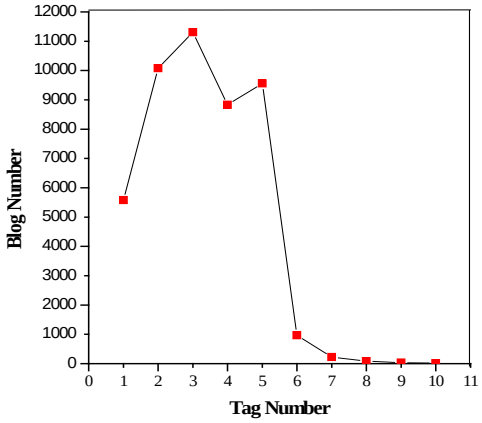

Fig. 1. Distribution of tag number of posts

3.2 Tag Number Distribution

Tags are important features for establishing classifiers in Section IV. The tag degree (Sun 2007) (tag degree of a blog is the number of tags attached to a blog) and tag rate (Song et al. 2010) (the percentage of tagged blog posts) demonstrate characteristics of tag features, and we can build a satisfactory classifier by taking full advantage of characteristics of tag features. Song et al. (2010) studied 25 Chinese websites and found that the average tag rate was 24.61%, and Sun et al. (2007) gave the tag number distribution in the BFE dataset (BlogFlux English, most blog posts were in English). But there are many differences between Chinese blogs and English blogs. A Chinese word always gives more information than an English word, so we assume that the number of tag of a Chinese blog is smaller than that of an English blog. The following Fig. 3 shows the tag number distribution of Chinese blog posts.

Fig. 1 shows that the number of the tags is limited and usually less than seven. For most blog posts, the tag number is concentrated from two to five. The average number of tags of a Chinese blog is 3.2, which is only half of the average of English Blog (Sun et al. 2007 found that each blog has an average of 6.3 tags with very few having more than 15), which confirms our above assumptions. The low tag rate and the small tag degree lead to the sparseness of tag features and the sparseness of tag features in Chinese blog posts is severer than that in English blog posts. If we use the sparse tag feature to establish a classifier, the performance of the classifier would not be satisfactory. Therefore, in Section 3.3, we propose an approach to propagate tag features to reduce the sparseness.

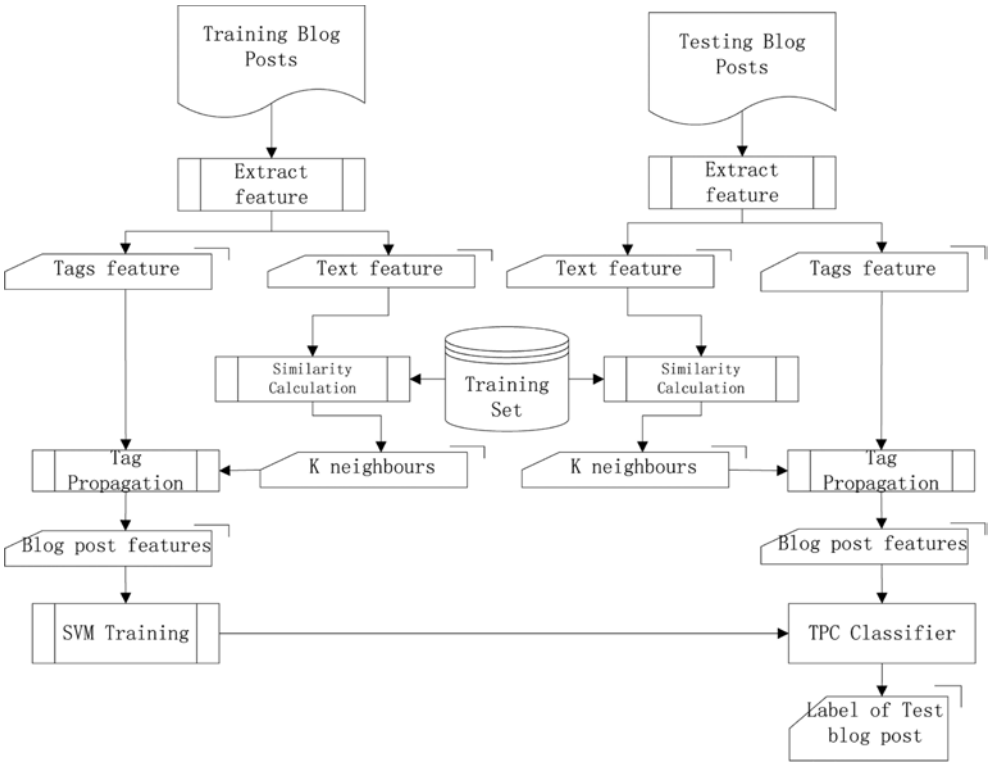

Fig. 2. Framework of tag-propagation-based blog post classification approach

3.3 Tag-propagation-based Feature Vector Building

The tag-propagation-based blog post classification approach is illustrated by the Fig. 2. For each training blog post, its tags are extracted to form original feature vectors. Then we find its nearest K neighbors from the training set and propagate the tags of the K blog posts to the training blog post by the calculated weight. A tag propagation classifier (TPC) is trained based on support vector machine (SVM) using the propagated feature vectors. For a test blog post, the same procedure is performed to construct its features, and the trained TPC is used to predict the category labels.

In our strategy, the tags are firstly extracted for the original features, and the final vectors used to train the classifier can be obtained according to the following steps.

1) Build tag feature vectors

This paper explains an experiment focusing mainly on tag features, so building tag feature vectors is the first step. In the experiment, N candidate tags that were extracted from the corpus are selected. In Section IV, we will present the process of selecting candidate tags. Using N tags, an N-dimensional vector in alphabetic order is built for each blog post in the training set. For every tag appearing in each blog post, we find its position in the vector and set its value to 1.

2) Find K neighbors based on content information

We choose M candidate words from the corpus. Based on the vector space model (VSM), we build an M-dimension vector to represent a blog post. Following this, for each blog post in the training set, we compute its similarity to each of the other blog posts in the training set by a cosine function to find its nearest K neighbors. Then, for each blog post in the testing set, we perform the same procedure to find its nearest K neighbors in the training set.

3) Tag propagation

For each blog post D in the training and testing set and its nearest K neighbors DN_1, …, DN_k (supposing that the i-th neighbor of D is DN_i), the blog post body similarity between D and DN_i is DS_i, and DN_i has tags DNT_{m1} ,…, DNT_{mi}. We propagate DNT_{ml}, …, DNT_{mi} to the N-dimension vector of D with weighted DS_i. If DNT_j is already in D, we do not perform tag propagation. Thus, given D and j, the weight of DNT_j is determined by the tag propagation function defined as follows.

$$weight(D, DNT_j) = \begin{cases} 1, & if\ D\ owns\ tag\ DNT_j \\ \sum_{i=1}^{k} \alpha_i \times DS_i \times own(DN_i, DNT_j), & others \end{cases} \quad (3\text{-}1)$$

where: $\alpha_i = \frac{1}{k} \sum_{j=1}^{k} DS_j$, $i = 1, 2, 3, \cdots$

$$own(DN_i, DNT_j) = \begin{cases} 1, & if\ DN_i\ owns\ tag\ DNT_j \\ 0, & others \end{cases} \quad i = 1, 2, 3, \ldots\ ,\ j = 1, 2, 3, \ldots \quad (3\text{-}2)$$

4 Experiments

4.1 Corpus

We carried out experiments on blogs collected from blog websites, such as IT51, CSDN, and ChinaUnix[1], from the years 2007 to 2012. All blog posts were organized in six categories (Web, Programming language, .Net technology, Linux Operating System, Database, and Mobile development) and each post contained content features and the following metadata: title, author, date, tags, category and so on. We crawled 47,000 blog posts in six categories, and most of them were in Chinese. As tags are our primary features for classification, we chose those with at least 1 tag to form our experiment corpus. In our experiment, we selected 12,000 blog posts as the experiment corpus with each category including 2000 blog posts, along with 1000 blog posts for training and 1000 for testing.

4.2 Preparation for Experiments

In the following experiment, we used different features to build classifiers to compare performance in blog classification problems. Therefore, we extracted features to be used to establish classifiers before a comparison experiment. Blog content features and tag features are main features used to establish classifiers, and the processing of extracting features will be concisely described in the following.

Content feature vectors: After word segmentation (different from English text, Chinese text needs word segmentation since there is no delimiter between adjacent Chinese words) using ICTCLAS[2] and filtering stop words in the experiment corpus, we gleaned a token list of each blog post. Then we selected features by DF (document frequency) (Yang et al. 19977; Mladenic et al. 1999; Li et al. 2007) and get 10,100 words as features finally. Using the VSM, we build a 10,100-dimension vector to represent blog content.

Tag feature vectors: Initially, the method proposed in Section 3.1 was used to extract tags from all blog posts in the corpus. After extracting the tags, we performed some tag cleaning work. Firstly, the stop words were removed from the candidate tags. Secondly, we combined some tags such as “linux”, “Linux”, and “LINUX”. Then we chose tag features using TF (term frequency) and obtained 3000 candidate tags. Finally, candidate tags were used to build 3000-dimension tag vectors to represent tag features of each blog post.

4.3 Classifier Selection

Although this paper focuses on tag feature propagation, in the comparison experiments, classifiers are built to explore the effectiveness of different features on the blog post classification task. Because a highly effective classifier is needed, the following paragraph compares the performance of different classifiers and selects the most effective one to conduct our assignment.

Various classifiers such as Naive Bayes, SVM, KNN, NN, and Rocchio could be adapted for blog post feature vectors generated from tag propagation. Yang et al. (1999) compared the five text classifiers above and showed that the SVM performs better. Generally speaking, SVM has the following advantages: on one hand, the SVM is more

[1] http://blog.51cto.com/
http://blog.csdn.net/
http://blog.chinaunix.net/
[2] http://ictclas.org/

effective in dealing with the situation when dimensionality of the text feature vector is very large; and on the other hand, text features have a high sensitivity to feature correlation, but the SVM is not sensitive to feature correlation, compared with other classifiers. For these reasons, we choose the SVM (Song 2004; Joachims 1998; Sum et al. 2002) to train the classifier with a linear kernel for classification of the blog post in this paper.

4.4 Find K Neighbors

In our approach, the nearest K neighbors are primary factors for propagating tags. In order to find the best K, we compute the classification precision, recall, and F1 by increasing K with. We randomly selected 200 blog posts for each category from testing set to tuning K. Fig. 3 shows the results.

We can see that Fig. 3 could be separated into 3 periods. At the beginning of the growth of K 1 to 5, the precision, recall, and F1 increase rapidly, which means that tag propagation has brought a blog post some category information. For K from 5 to 20, the growth of the 3 evaluation criteria slows down by degrees and reaches a peak value when K=20. Then the growth reaches to a smooth state and fluctuates slightly, which means that a further increase of K will not bring additional information.

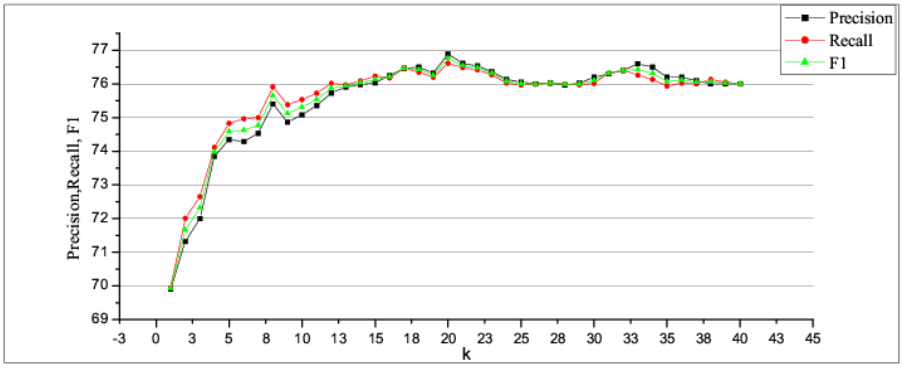

Fig. 3 .Classification performance with the growth of K

4.5 Experiments

In our experiments, we compared our approach to several baselines that have different features or use different similarity calculation methods. From Section 4.3, we know that SVM performs better, so all the comparative experiments used SVM to establish the classifier.

4.5.1 Performance Comparison and Analysis

In order to observe the effectiveness of our approach, we compared the performance of several feature extraction methods based on the SVM classifier.

BFC (classifier trained by body features only): We build the body feature in “find K neighbors” in Section 3.1, and obtained a 10,100-dimensional vector. We used the 10,100-dimensional body features as training features, and gained a classifier (BFC).

TFC (classifier trained by tag features only): In section 4.2, we have obtained 3000-dimensional tag vector by the tag extraction process, and trained a classifier (TFC) by

using the 3000-dimensional tag features.

BTFC (classifier trained by combining features): We use the simplest method to combine body features and tag features. We concatenated the tag feature vector to a text feature vector and got 13,100-dimensional hybrid features. A BTFC was built using the hybrid features as training features.

TPC (classifier trained by propagated tag features): We propagated tag features by the tag propagation algorithm proposed in Section 3.3, then trained a TPC by using the propagated tags as training features.

We performed experiments to explore the performance of classifiers trained by using features: BFC, TFC, BTFC, and TPC. In tag propagation, the similarity was calculated by cosine similarity, and the word weight was computed by TF*IDF score. The results are shown in Table 1.

Classifier Feature	Precision (%)	Recall (%)	F1 (%)
BFC	64.98	61.70	63.02
TFC	71.39	70.49	70.94
BTFC	71.79	72.0	71.81
TPC	76.89	76.62	**76.75**

Table 1 Accuracy of different classifiers using different features (K = 20)

Table 1 shows that a classifier trained based on tag features is better than that trained based on content features, which means using tag features was more effective than using content features in blog post classification (Brooks et al. 2006). To measure the effect of tag propagation, we applied the tag propagation method to propagate all tags of blog posts including training blogs and testing blogs. In Table 1, the last item reports the performance of tag propagation. With the tag propagation, the F1 was improved by 13.73%, with an improvement of precision of 11.91% and recall of 14.92% over BC classifier (using body feature only), which indicates our approach is an effective method to classify blog posts.

4.5.2 Influence of similarity calculation methods

Although our approach focuses on tag propagation, text similarity is a major factor in the tag propagation process. As is known from Section 3.3, text similarity is an important part of formula (4) and determines the performance of tag propagation. Therefore, we conducted the following two sub-experiments to investigate the performance of our approach under different similarity calculation methods.

Cosine similarity: Cosine similarity is one of the most popular similarity measurements. Given two documents X and Y, their cosine similarity is

$$CosSim(X,Y) = \frac{X \cdot Y}{\|X\| \times \|Y\|} \quad (4\text{-}1)$$

Each dimension represents a term with its weight in the document, which is non-negative.

Jaccard similarity coefficient: The Jaccard similarity coefficient is always used to calculate the similarity of two sets that use binary weight to represent feature attributes. If we use the binary score as the weight of words in the VSM, the Jaccard similarity coefficient of two blogs will be calculated by the following formula.

$$JaccardSim(X,Y) = \frac{X \cap Y}{X \cup Y} \quad (4\text{-}2)$$

Here, *X, Y* represents the word sets with a binary score of blogs.

Correlation coefficient: The correlation coefficient is known as Pearson correlation coefficient in statistics. It can evaluate the correlation degree between two variables. We can calculate the correlation coefficient of two blog vectors by using the following expression.

$$RSim(X,Y) = \frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sqrt{\left\|X-\overline{X}\right\|^2 \left\|Y-\overline{Y}\right\|^2}} \quad (4\text{-}3)$$

In order to explore the effect of similarity calculation methods to tag propagation, we listed the performance achieved using different methods of similarity calculation in Table 2 and Table 3.

Similarity Methods	Precision (%)	Recall (%)	F1 (%)
Cosine	76.89	76.62	76.75
Jaccard	×	×	×
Correlation coefficient	76.92	76.67	76.79

Table 2 Tag Propagation using three different similarity calculation methods, and the word weight was computed by *TF*IDF* score in the similarity calculation process. (K = 20)

Similarity Methods	Precision (%)	Recall (%)	F1 (%)
Cosine	79.63	79.23	79.42
Jaccard	71.39	73.0	72.18
Correlation coefficient	79.64	79.33	79.48

Table 3 Tag Propagation using three different similarity calculation methods, and the word weight was computed by *Binary* score in the similarity calculation process. (K = 20)

Table 2 and Table 3 report the performance of tag propagation using different similarity calculation methods and different representation methods of word weight. In Table 2, word weight was computed by TF * IDF. In Table 3, it used binary to measure word weight. From Table 2 and Table 3, we can find that the cosine and correlation coefficient had better performance than Jaccard in tag propagation methods. Tag propagation based on a binary value was more effective than that based on TF*IDF on the cosine and correlation coefficient. When using BFC classifier (using body feature only) as a baseline, the F1 score maximum improvement was 16.46%, with improvement of precision by 14.66% and recall by 17.63%.

5. Conclusions

In this paper, we presented a blog post classification approach based on the tag propagation strategy. The contributions of this paper can be summarized as follows:

(1) with tags from the content-similar blogs, tag propagation approach complements information for the tag features and makes binary feature vectors become real-value ones. Thus, the feature sparseness is reduced and the performance of the blog post classifier is improved;

(2) compared with traditional strategies that mainly take content word features and consider the tags as supplementary, our method enhances the effect of classifiers taking

only tags as features;

(3) according to the results of comparative experiments, the method of similarity calculation is very important in our approach. An appropriate similarity calculation method can propagate tags more effectively and improve the performance of classifiers.

In the future, our work will be carried out along the following directions: firstly, we will explore more reasonable propagating algorithms to improve the accuracy of the classification methods; secondly, we will investigate similarity calculation methods that can compute similarity more accurately; thirdly, the principle of tag propagation will be extended to the solutions of other related problems.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grants No.61103149, No.61100094 and No.61300114. The authors are grateful to the anonymous reviewers for their constructive comments. Special thanks to Haifeng Hu, Yu Jiang, Chengjie Sun and Ming Liu for insightful suggestions.

References

Sun, A., Suryanto, M. A., & Liu, Y. (2007). Blog classification using tags: An empirical study. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pp. 307-316.

Tsai, F. S. (2011). A tag-topic model for blog mining. *Expert Systems with Applications*, vol. *38*, no. 5, pp. 5330-5335.

Sood, S., Owsley, S., Hammond, K. J., & Birnbaum, L. (2007, March). TagAssist: Automatic Tag Suggestion for Blog Posts. In ICWSM.

Mathes, A. (2004). Folksonomies-cooperative classification and communication through shared metadata. *Computer Mediated Communication*, vol. *47*, no. 10,pp. 1-13.

Macgregor, G., & McCulloch, E. (2006). Collaborative tagging as a knowledge organisation and resource discovery tool. *Library review*, vol. *55 no.*5, pp. 291-300.

Marlow, C., Naaman, M., Boyd, D., & Davis, M. (2006, August). HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pp. 31-40.

Berendt, B., & Hanser, C. (2007, March). Tags are not metadata, but" just more content"-to some people. In *ICWSM*.

Brooks, C. H., & Montanez, N. (2006, March). An Analysis of the Effectiveness of Tagging in Blogs. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, vol. 6, pp. 9-14.

Brooks, C. H., & Montanez, N. (2006, May). Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th international conference on World Wide Web*, pp. 625-632.

Veeramachaneni, C. H. P. A. S., Hayes, C., & Avesani, P. (2007). An analysis of the use of tags in a blog recommender system. In *IJCAI'07*, pp. 2772-2777.

Wang G, Hoiem D, and Forsyth D. Building Text Features for Object Image Classifications. In proceedings of CVPR 2009.

Vargas-Vera, M., Domingue, J., Kalfoglou, Y., Motta, E., & Buckingham Shum, S. (2001). Template-driven information extraction for populating ontologies.

Collier R. Automatic Template Creation for Information Extraction. PhD thesis, Department of Computer Science, University of Sheffield, UK, September 1996.

Yang, Y., & Pedersen, J. O. (1997, July). A comparative study on feature selection in text categorization. In *ICML*, vol. 97, pp. 412-420.

Mladenic, D., & Grobelnik, M. (1999, June). Feature selection for unbalanced class distribution and naive bayes. In *ICML*, vol. 99, pp. 258-267.

Li X, Yan H, and Wang J. Search Engine: Principle, Technology and Systems. Science Press, 2007, pp. 209-210.

Yang, Y., & Liu, X. (1999, August). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 42-49.

Song F. Studies on Some Essential Problems in Automatic Text Categorization. Jiangsu: Nanjing University of Science and Technology, doctoral dissertation, February 2004.

Joachims, T. (1998). *Text categorization with support vector machines: Learning with many relevant features*, pp. 137-142.

Sun, A., Lim, E. P., & Ng, W. K. (2002, November). Web classification using support vector machine. In *Proceedings of the 4th international workshop on Web information and data management*, pp. 96-99.

Song H, Li L, and Liu D. Research on Chinese Blog Tags and Recommendation Model. YWCL2010, pp. 310-316.