Social Tag Prediction Base on Supervised Ranking Model

Hao Cao, Maoqiang Xie, Lian Xue, Chunhua Liu, Fei Teng, and Yalou Huang

College of Software, Nankai University, Tianjin, P.R.China
{caohao, xuelianlotus, hytjfxk, nktengfei}@mail.nankai.edu.cn
{xiemq,huangyl }@nankai.edu.cn

Abstract. Recently, social tag recommendation has gained more attention in web research, and many approaches were proposed, which can be classified into two types: rule-based and classification-based approaches. However, too much expert experience and manual work are needed in rule-based approaches, and its generalization is limited. Additionally, there are some essential barriers in classification-based approaches, since tag recommendation is transformed into a multi-classes classification problem, such as tag collection is not fixed. Different from them, ranking model is more suitable, in which supervised learning can be used. In additions, the whole tag recommendation task can be divided into 4 subtasks according to the existence of users and resources. In different subtasks, different features are constructed, in order that existed information can be used sufficiently. The experimental results show that the proposed supervised ranking model performs well on the training and test dataset of RSDC 2008 recovered by ourselves.

1 Introduction

Tag is a new form to index web resources, which help users to categorize and share the resources, and later search them. Also, the tags assigned by specified user reveal the user's interests, therefore, according to the tags user have already tagged, someone can find other users who have the similar interests, as well as similar interesting resources. Therefore, it is widely used in social network such as Bibsonmy, Del.icio.us, Last.fm , etc.

A tag recommendation system can suggest someone a few tags to specified web resource, thus it can save the user time and effort when them mark up resources. Further, the recommended tags and existing tags can be used to predict the profile of the user and the interesting to the web resource, for example, to predict what they like and dislike. The research of tag recommendation is also very suggestive for other applications, such as online advertisement. In the field of online advertisement, we can predict what advertisement the browser might be interested in with the help of the surrounding text and his browsing history.

Recently, social tag recommendation has gained more attention in web research. It has been a hot issue for both industry and research area. For example, tag recommendation is one of the tasks in ECML RSDC's 08. Now, in ECML

PKDD 09, tag recommendation has become the exclusive task. However, the performance of tag recommendation is not good enough to be widely used, more research work is needed and progress is essential for the practical use of tag recommendation in commercial system. In this paper, supervise ranking model is applied to tackle tag recommendation problem, and good result is achieved on test data.

The rest of paper is organized as follows: Section 2 lists the previous work on tag recommendation. Section 3 gives a description of supervised ranking model. Section 4 lists our experiment settings, experiment procedure and our analysis of the results on recovered 08's dataset. The model's performance on 09's dataset is presented in Section 5. Section 6 summarizes our work.

2 Previous Work

Much research work has been done for tag recommendation, most of which can be categorized into two types, one is rule-based, the other is classification-based.

Rule based approach is used by many researchers. Lipczak [1] proposed a three-step tag recommendation system in their paper : Basic tags are extracted from the resource title. In the next step, the set of potential recommendations is extended by related tags proposed by a lexicon based on co-occurrences of tags within resource's posts. Finally, tags are filtered by the user's personomy - a set of tags previously used by the user. Tatu, et al. [2]used document and user models derived from the textual content associated with URLs and publications by social bookmarking tool users, the textual information includes information present in a URL's title, a user's description of a document, or a bibtex field associated with a scientific publication, they used natural language understanding approach for producing tag recommendations, such as extraction of concepts, extraction of conflated tags which group tags to semantically related groups. However, too much expert experience and manual work are needed in rule-based approaches, and its generalization is limited.

Classification-based approach is also used for the tag recommendation task. Katakis et al. [3] tried to model the automated tag suggestion problem as a multilabel text classification task. Heymann et al. [4] predicted tags based on page text, anchor text, surrounding hosts, and other tags applied to the URL. They found an entropy-based metric which captures the generality of a particular tag and informs an analysis of how well that tag can be predicted. They also found that tag-based association rules can produce very high-precision predictions as well as giving deeper understanding into the relationships between tags. Their results have implications for both the study of tagging systems as potential information retrieval tools, and for the design of such systems. However , the application of classification does not suggest a good solution to the tag prediction problem: first, the tag space is fixed , all the resource can be categorized to the existed tags only, also, the amount of tags number could be very large, the traditional classification model would be rather low efficient.

Collaborative filtering is a commonly used technical for user-oriented task. Many researchers tried collaborative filtering in tag recommendation. Gilad Mishne [5] used collaborative approach to automated tag assignment for weblog posts. Robert Jaschke, et al [6] evaluated and compared user-based collaborative filtering and graph-based recommender, the result shows that both of these two methods provide better results than non-personalized baseline method, especially the graph-based recommender outperforms existing methods considerably.

Adriana Budura et al. [7] used neighborhood based tag recommendation, which make use of content similarity. Principle and simple score approach is used to select the candidate tags, however, in our paper, machine learning method is used, a ranking model is learned automatically, then the candidate tags are ranked and top-ranked tags are suggested as recommending tags.

3 Supervised Ranking Model for Tag Recommendation

3.1 Problem Statement

The tag recommendation problem can be described as follows:

For a given post P whose user is U and resource is R, a set of tags are suggested as tags for the post. Here we denote post as P, tag as T, resource as R, user as U.

A possible and most nature way to solve the tag recommendation problem is as follows: First, a set of candidate tags are selected for the post, and then tags which are most likely to be the tags for the post are selected as recommending tags. The commonly used approach to choose the tags is rule-based and classification-based methods, but both of them have defects: rule-based approach relies on expert experience and manual efforts to set up the rules and tuning the parameters; classification-based is restrict to the fix of tag space and is inefficient when it is treated as a multi-label problem. In this paper, tag recommendation is conveyed to a problem of ranking candidate tags. A ranking model is constructed to ensure tags that are most likely to be post's tags rank higher than tags that are not. Supervised learning model is used to construct the ranking model satisfying the restriction. Ranking-SVM model is the most frequently used supervised ranking model and is proofed to be a successful model, so it is used as our supervised ranking model in the experiments. All the candidate tags for one post are grouped as a ranking group and the top-ranked candidate tags are selected as recommendation tags.

3.2 Introduction to Ranking SVM

Here we briefly describe the Ranking Support Vector Machine(Ranking SVM) model for tag recommendation.

Assume that $\mathcal{X} \in \Re^m$ is the input feature space which represents feature of a candidate tag given a user and resource, and m denotes the feature number. $\mathcal{Y} = \{0, 1\}$ is the output rank space which is represented by the labels, and 1 represents the tag is labeled by user, and 0 is not. $(x, y) \in \mathcal{X} \times \mathcal{Y}$ denotes feature and label as the training instance.

Given a training set with tags $T = \{t_1, t_2, ..., t_n\}$, for each tag t_i there would be a $\{x, y\}$ associated with it, the whole training set could be formulate as $S = \{x_i, y_i\}_{i=1}^{N}$, where N represents the number of all tags.

In Ranking SVM [8], ranking model f is a linear function represented by $\langle w, x\rangle$, where w is the weight vector and $\langle\cdot,\cdot\rangle$ denotes the inner product. In RSVM we need to construct a new training set S' according to the original training set $S = \{x_i, y_i\}_{i=1}^{N}$. For every $y_i \neq y_j$ in S, construct $(x_i - x_j, z_{ij})$ and add it into S', where $z_{ij} = +1$ if $y_i \succ y_j$, and otherwise -1. Here $\succ$ denotes the preference relationship, for example, $y = 1$ is preferred to $y = 0$. For denotation consistency, we denotes S' as $\{x_i^1 - x_i^2, z_i\}_{i=1}^{D}$. The final model is formalized as the following Quadratic Programming problem:

$$\begin{aligned} & min_{w,\xi_i} \frac{1}{2C}\|w\|^2 + \sum_{i=1}^{D} \xi_i \\ s.t. \quad & \xi_i > 0, z_i\langle w, x_i^1 - x_i^2\rangle \geq 1 - \xi_i \end{aligned} \tag{1}$$

And (1) could be solved using existing Quadratic Programming methods.

Figure 1 is an example of ranking SVM model.

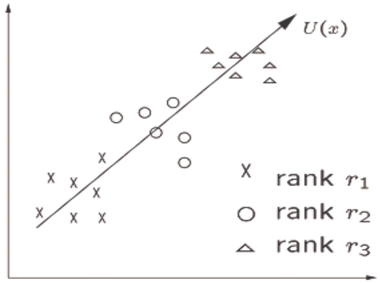

Fig. 1. Example of ranking SVM model

The ranking SVM model convey the problem of ranking into binary classification problem: for each objects to be ranked, the model compare it with all other objects in the same ranking group. For n objects, the model compares the objects C_n^2 times, and then outputs the ranking result.This is the advantage over classification model: in classification model, the existence of other candidate tags is not being considered, but in ranking model, the existence of other candidate tags is taken into consideration.

3.3 Ranking Process

For any post P_{ij} in test dataset, we denote collection of all candidate tags for post P_{ij} as $CT\{P_{ij}\}$ and $CT_k(k = 1, 2, ..., n)$ as the k-th candidate tag for the post P_{ij} , $CT\{P_{ij}\} = \{CT_1, CT_2, ..., CT_n\}$. The ranking model ranks the candidate tags to $\{{CT_1}', {CT_2}', ..., {CT_n}'\}$ from top to bottom. Then top-k tags are selected as prediction of the tags of post P_{ij}. Table 1 shows the steps to rank the candidate tags.

Table 1. Algorithm of rank the candidate tags

```
Input: candidate tags $\{CT_1, CT_2, ..., CT_n\}$
Output: top-k tags $\{CT_1', CT_2', ..., CT_k'\}$

1.  Extract feature $x = \{x_i\}(i = 1, 2, ..., n)$ for a sequence
    of candidate tags $CT\{P_{ij}\} = \{CT_1, CT_2, ..., CT_n\}$.
2.  Rank the features using the learned ranking model as
    $\{CT_1', CT_2', ..., CT_n'\}$.
3.  select top-k tags $\{CT_1', CT_2', ..., CT_k'\}$ as recommending tags.
```

Also, the number of recommended tags affects the performance of the system. For example, if the actual number of tags for post whose content_id=123456 is 3, a loss of precision is suffered when 4 tags are recommend to the user. So a proper number of tags to recommend should be found. The number used in our experiment is half the number of all candidate tags. If the number is bigger than 5, we cut them into 5, that means we recommend 5 tags at most.

3.4 Training Process

For all the post in the test dataset, candidate tags $CT\{P_{ij}\}$ for each post P_{ij} are extracted. Then they are grouped by the post, and features are extracted for each of them in the post content. For those $CT_k \in T\{P_{ij}\}$, we label them '1', else label them with '0'. Then we use SVM-light tool to train a ranking-SVM model. When predicting the tags of the post in test dataset, the model learned on the training dataset is applied to rank the candidate tags, and top ranked tags are selected as recommending tags.

4 Experiments on 08's recovered dataset

4.1 Experiment settings

2008's dataset recovery In order to compare our experiments' performance with that of the 08's teams, we try to get the 08's dataset (both training and test data) and test our model's performance on the recovered dataset. Though the 08's test data can be downloaded from the web, we found that user IDs have been changed between the datasets. However, the content_id field in 08's test data is consistent with 09's data, so we try to recover the 08's dataset on the 09's dataset using the content_id field and date time field. The 08's real training data and test data are subset of 09’s data, so it is possible to recover 08’s data on 09’s data. After observing 08’s real test data, we found that all posts in 08’s test data are between Mar. 31, 2008 and May. 15, 2009, so we use the posts during this period on 09’s training data as recovered 08’s test data and posts before Mar. 31, 2008 as our recovered 08’s training data. There are still slight difference between our recovered data and the 08’s real data. We assume that the difference won’t affect our performance seriously, so the result is comparable with 08’s results.

Some statistics have been made on our recovered 08’s dataset. Table 2 shows the statistics of posts on this recovered dataset. Table 3 shows the statistics of posts according to the existence of their user and resource in the recovered training data. In following part in section 4, the training data refers to the recovered training data, the test data refers to the recovered test data.

Table 2. Statistics of posts on recovered 08’s dataset

Post in recovered training data	234,134	BOOKMARK	184,655
		BIBTEX	49,479
Post in recovered test data	63,192	BOOKMARK	20,647
		BIBTEX	42,545

Table 3. Statistics of posts according to their user and resource status

Users in recovered test data appear in recovered training data	265
Users in recovered test data do not appear in recovered training data	225
Resources in recovered test data appear in recovered training data	1230
Resources in recovered test data do not appear in recovered training data	61970

Data format description The dataset used in experiments is released by ECML. The data consists of three tables: TAS table, BOOKMARK table and BIBTEX table.

Table 4 is a description of the fields of the three tables. Only the fields we used in experiments are listed in the table.

Data preprocess Firstly, the terms are converted into lowercase. Then the stop words are removed, such as ”a, the, is, an”, these terms are not likely to be the tags of the post. Finally, the punctuations as ’:’, ’,’, etc are removed. Latex symbols such as ’{’ and ’}’ is also removed using regular expressions.

Table 5 shows example results of data preprocess.

4.2 Post Division

It can be observed from data distribution that some users of posts exist in the training data (54%) and some do not exist in the training data (46%). Also

Table 4. Data fields of TAS, BOOKMARK and BIBTEX

Table name	Fields name
TAS	user, tag, content_id, content_type, date
BOOKMARK	content_id (matches tas.content_id) ,url description ,extended ,description ,date ,bibtex
BIBTEX	content_id (matches tas.content_id) ,simhash1 (hash for duplicate detection among users) ,title

Table 5. Example results of data preprocess

Before data preprocess	After data preprocess
Ben Mezrich: the telling of a true story	ben mezrich telling true story
{XQ}uery 1.0: An {XML} Query Language, {W3C} Working Draft	xquery 1.0 xml query language w3c working draft

some resources of posts exist in the training data (2%) and others do not exist in the training data (98%).

In the analysis above, we divide the posts in test dataset into two categories according to the existence of their users in the training data: existed user posts, non-existed user posts. Also, the posts in test dataset can be divided into two categories according to the existence of their resource in the training data: existed resource posts, non-existed resource posts.

The posts can be divided into four different categories according to their user status and resource status in the training data: existed user existed resource post, existed user non-existed resource post, non-existed user existed resource post, non-existed user non-existed resource post.

We denote symbols as shown in Table 6 to simplify the language.

Table 7 and Table 8 show statistics after our post division on our recovered 08's data.

Table 6. Simplified symbols

EUER post	Existed user existed resource post
EUNR post	Existed user non-existed resource post
NUER post	Non-existed user existed resource post
NUNR post	Non-existed user non-existed resource post

It can be observed from statistics that not every category of posts occupies the same ratio of the posts. In BOOKMARK, EUNR posts occupied about 82.80% of all BOOKMARK posts. In BIBTEX, NUNR posts occupied about 93.43% of all BIBTEX posts. In order to promote our model's performance on

Table 7. Distribution of different categories of BOOKMARK posts in test dataset

Category	Posts number	ratio
EUER post	621	3.01%
EUNR post	17099	82.80%
NUER post	346	1.68%
NUNR post	2585	12.52%

Table 8. Distribution of different categories of BIBTEX posts in test dataset

Category	Posts number	ratio
EUER post	164	0.39%
EUNR post	2532	5.95%
NUER post	99	0.23%
NUNR post	39754	93.43%

the test dataset, we should focus on those data which occupy high proportion of the posts, that is: EUNR posts of BOOKMARK and NUNR posts in BIBTEX.

After data division, the following steps are carried out for our tag recommendation task.

1. Extract candidate tags by different methods according to the category of post.

2. Rank the candidate tags, and select top ranked tags as recommendation tags.

4.3 Candidate tags extraction

According to the statistics of the sources of the tags on the dataset, we can find that tags can be retrieved from three sources mainly: 1.The content information of the post, such as 'description' field in BOOKMARK and 'title' field in BIBTEX. 2. $T\{R_j\}$: The tags being assigned to the same resource previously. 3.$T\{U_i\}$: The tags assigned by the same user previously. Statistics of tags from different sources for BOOKMARK and BIBTEX posts are listed in Table 9 and Table 10.

Table 9. Statistics of the tags from 3 sources of BOOKMARK Post

Total tags	56267
Tags from terms of description	5253
Tags from terms of URL	1353
Tags from user's previous tags	29672

Table 10. Statistics of the tags from 3 sources of BIBTEX Post

Total tags	95782
Tags from terms of title	41801
Tags from terms of URL	547
Tags from user's previous tags	5377

The four different categories of test dataset have different characters, for example, we can explore the tags assigned by user previously and the tags assigned to the resource previously for EUER posts. But for NUNR posts, we lack this information. So we should explore different features for the four different categories of posts individually, in order that existed information can be used sufficiently. In the following part, while using the supervised ranking model, we train four models to handle these four categories of posts individually.

The candidate tags extraction strategies for different categories of posts:

For EUER post and NUER post, $CT\{P_{ij}\} = \{$ terms in post $(P_{ij}) \bigcup T\{R_j\}\}$.

For EUNR post and NUNR post, $CT\{P_{ij}\} = \{$ terms in post $(P_{ij})\}$.

We denote the candidate tags for post whose user_id=i and resource is j as $CT\{P_{ij}\}$. $\{$ terms in post $(P_{ij})\}$ denotes the remaining set of words after trimming and removing of the stop words in the text information of post P_{ij}.

Notice should be paid here that we do not take $T\{U_i\}$ (the user's pervious tags) as candidate tags because we find the tags are too massive. When they are added, the precision of the system drops down and the F-1 value on the whole dataset also declines dramatically. However, in the ranking procedure, we will use $T\{U_i\}$ as one of the features in SVM model to rank the candidate tags.

4.4 SVM Features construction

While using SVM, we select features that discern high ranked tags and low ranked tags well and add the features according to our experience. For example, the term frequency in the post content: those words which have high term frequency within the post content tend to rank higher than those which have low term frequency. Also, whether the candidate words have been used as tags for other post in the training data is an excellent feature.

Table 11 is a brief description of features of ranking SVM model for BOOKMARK posts. The features for BIBTEX posts are almost the same except for the different data fields:

4.5 Analysis of Model

Table 12 and Table 13 show the results of our supervised Ranking SVM model on the recovered 08's data.

Combing different types and category of data together, we can get the overall performance on the recovered 08's test data, as shown in Table 14.

Table 11. Some of the features for ranking SVM model for BOOKMARK

Feature1	Candidate tag's TF (term frequency) in post's description terms.
Feature2	Candidate tag's TF in post's URL terms.
Feature3	Candidate tag's TF in post's extended description terms.
Feature4	Candidate tag's TF in $T\{R_j\}$ (tags assigned to the post of the same URL in the training data).
Feature5	Candidate tag's TF in $T\{U_i\}$ (tags assigned previously by user in the training data.)
Feature6	Times of candidate tag being assigned as a tag in the training data.

Table 12. Individual and overall Performance on BOOKMARK posts

Post category	Recall	Precision	F1-value	ratio
EUER Post	0.369699	0.394973	0.381918	3.01%
EUNR Post	0.046591	0.053739	0.04991	82.80%
NUER Post	0.160883	0.255652	0.197487	1.68%
NUNR Post	0.069158	0.106366	0.083819	12.52%
overall-performance on BOOKMARK	0.061067	0.073997	0.066633	

Table 13. Individual and overall Performance on BIBTEX posts

Post category	Recall	Precision	F1-value	ratio
EUER Post	0.4219356	0.3472393	0.3809605	0.39%
EUNR Post	0.2250226	0.1628605	0.1889605	5.95%
NUER Post	0.5667162	0.3715986	0.4488706	0.23%
NUNR Post	0.3561221	0.1603686	0.2211494	93.43%
overall-performance on BIBTEX	0.349063	0.161732	0.220381	

Table 14. Overall performance on test dataset using ranking SVM model

Recall	Precision	F1-value
0.153	0.185	0.167

The F1-value is 0.167, less than the F1-value 0.193 of the team ranked first in 08's competition.

It can be observed from the results that the performance of the model is poor on EUNR posts, which occupied most of the BOOKMARK posts. However, the model performs well on EUER posts. When comparing the two types of data, we find that the only difference is that the candidate tags of EUER posts are not only come from the post content but also from the tags of the same resource in the training data, however, the candidate tags for EUNR posts come from post content only. In order to overcome the weakness of lacking candidate tags, we relax restriction on the definition of the same resource. For those posts whose resources have not appeared in the training data, the role of the same post is substituted by the similar post. This method is based on the assumption that users tend to tag the similar posts with the same tags.

We try to use post content similarity to measure the similarity of posts. For those EUNR posts, which have no same resources in the training data, we add the tags of those posts whose content similarity with the current post content is above a certain threshold to the candidate tags set of the post.

4.6 Post content similarity based KNN model

For EUNR post, the candidate tags come from text of the post content only, that is $CT\{P_{ij}\} = \{$ terms in post $(P_{ij})\}$. We attribute the poor performance of the model on such kind of data to the sparse of candidate tags. So we use content similarity to expand the candidate tags set. For any EUNR post P_{ij}, we set a similarity threshold t, and find in the training dataset content P_{mn}, whose $sim(text(P_{ij}), text(p_{mn}) > t)$. Then the tags of post P_{mn} are added to the candidate tags ofP_{ij}: $CT\{P_{ij}\} = \{$ terms in post $(P_{ij})\} \bigcup T\{P_{mn}\}$.

Post content P_{ij} and P_{mn} are mapped into vector space:

$text(P_{ij}) = \{W_1, W_2, ..., W_n\}$, $text(P_{mn}) = \{{W_1}', {W_2}', ..., {W_n}'\}$,Then we use vector space model to calculate the similarity between two posts P_{ij} and P_{mn}.

$$sim(text(P_{ij}), text(P_{mn})) = \frac{text(P_{ij}) * text(P_{mn})}{|text(P_{ij})| * |text(P_{mn})|} \tag{2}$$

W_i means the weigh of word i in the content. The simplest way to define W_i is as following:$W_i = 0$,word i in post content, $W_i \neq 0$,word i not in post content.

In our experiment, we define the W_i as TF(Term Frequency) multiply IDF (Inverted document frequency) :$W_i = TF_i * IDF_i$.We applied open source software Lucene to calculate the similarity of two content , the scoring function of Lucene is a derivation of vector space model formula using TF/IDF weighing schema.

The modification of threshold value T and the corresponding performance on EUNR content in BOOKMARK are shown in Figure 2.

It can be observed that the value of recall, precision and F1 value reach highest when threshold T=0.5. So, in the further experiment settings, we set threshold value T to 0.5.

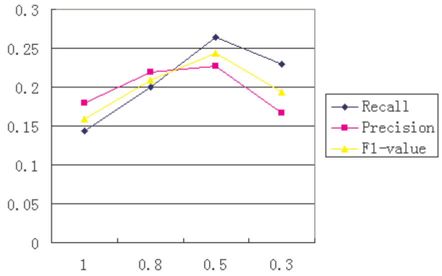

Fig. 2. KNN performance on various threshold t on BOOKMARK EUNR posts, k=5

However, we find that the application of content similarity based KNN model works for BOOKMARK posts but not for BIBTEX posts. After investigation, we attribute it to the uneven distribution of the dataset in training datasets and test datasets. In training datasets, the number of BOOKMARK posts is 184,655 and the number of BIBTEX posts is 20,647. But in test dataset, the number of BOOKMARK post is 20,647 and the number of BIBTEX post is 49,479, it is easy for 20,647 BOOKMARK posts to find similar posts in 184,655 BOOKMARK posts, but difficult for 42,545 BIBTEX posts in only 20,647 posts. So this method is especially useful for BOOKMARK posts but not for BIBTEX posts.

After applying content similarity based KNN model on BOOKMARK EUNR posts, the performance on overall test dataset is as listed in Table 15.

Table 15. Overall performance on test dataset adding content similarity based KNN model

Recall	Precision	F1-value
0.323828	0.200926	0.238803

The F1-value is 0.238, higher than the F1-value 0.193 of the team ranked first in 08s competition.

5 Experiment on 09's dataset

5.1 Statistics of 09's dataset

Table 16 and Table 17 show the distribution of different categories of posts on 09's dataset after data division according to the existence of their user and resource in the training data. In our experiment settings on 09's test data, clean-dump dataset is used as training dataset in Task 1, Post-core dataset is used as training dataset in Task 2.

Table 16. Different categories of BOOKMARK posts in 09s test dataset for Task 1

Category	Posts number	ratio
EUER Post	821	4.86%
EUNR Post	10622	62.86%
NUER Post	872	5.16%
NUNR Post	4583	27.12%

Table 17. Different categories of BIBTEX posts in 09s test dataset for Task 1

Category	Posts number	ratio
EUER Post	365	1.40%
EUNR Post	9287	35.71%
NUER Post	591	2.27%
NUNR Post	15761	60.61%

It can be observed from the statistics of the distribution of categories in 09's test data for Task 1 agrees with the recovered 08's dataset: EUER posts occupied most of the BOOKMARK post and NUNR post occupied large proportion of BIBTEX posts, so we can expect our model a good result on such data. The whole posts in 09's test dataset for Task2 can be classified to EUER posts. Since the good performance of our model on EUER posts, we can also expect a good result on task 2.

Eight different models are trained on 09's clean-dump training data and applied in 09's test data for Task 1. For Task 2, we apply the BOOKMARK EUER post model and the BIBTEX EUER post model trained on 09's post-core dataset.

5.2 Experiment results on 09's test dataset

The performance on the whole 09's test data of both task 1 and task 2 is shown in Table 18.

Table 18. Performance on 09's dataset @5

Task No.	Submission ID	Precision	Recall	F1-value
1	67797	0.162478	0.146582	0.154121
2	13651	0.31622	0.222065	0.260908

6 Conclusion

In this paper, we briefly describe an approach utilizes supervised ranking model for tag recommendations. Our tag prediction contains three steps. First, posts are divided into four categories according to the existence of the user and the resource in the training data and then candidate tags are extracted for the different categories with different strategies. Second, features are decided according to categories. Then we rank the candidate tags, using the supervised ranking model, and pick the top tags as recommendation tags.

For the existed user non-existed resource post, we use post content similarity based KNN model to expand the candidate tags set. Performance of this experiment for the corresponding module is promoted after adding this model on 08's dataset. Our tag recommendation system is generated from the combination of these two models and applied to the 09's tags recommendation task 1 and task 2.

Acknowledgement

Thanks to Zhen Liao for his helpful discussions and suggestions for this paper.

This paper is supported by the National Natural Science Foundation of China under the grant 60673009 and China National Hanban under the grant 2007-433.

References

1. Marek Lipczak, Tag Recommendation for Folksonomies Oriented towards Individual Users, *ECML 2008*
2. Marta Tatu, Munirathnam Srikanth, and Thomas D'Silva, RSDC'08: Tag Recommendations using Bookmark Content, *Proceedings of ECML PKDD Discovery Challenge 2008 (RSDC 2008)*
3. Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas, Multilabel Text Classifcation for Automated Tag Suggestion, *Proceedings of ECML PKDD Discovery Challenge 2008 (RSDC 2008)*
4. Paul Heymann, Daniel Ramage, and Hector Garcia-Molina, Social Tag Prediction, SIGIR'08, July 20–24, 2008, Singapore.pages 531∼538
5. Gilad Mishne,AutoTag: A Collaborative Approach to Automated Tag Assignment for Weblog Posts. *WWW 2006,* May 22–26, 2006, Edinburgh, Scotland. pages 953 954
6. Robert J¨aschke, LeandroMarinho, Andreas Hotho,Lars Schmidt-Thieme, and Gerd Stumme, Tag Recommendations in Folksonomies J.N. Kok et al. (Eds.): *PKDD 2007,* LNAI 4702, pp. 506–514, 2007.
7. Adriana Budura, Sebastian Michel, Philippe Cudre-Mauroux, and Karl Aberer, Neighborhood-based Tag Prediction, *6th Annual European Semantic Web Conference (ESWC2009)*
8. R Herbrich, T Graepel, K Obermayer ,Large margin rank boundaries for ordinal regression, textitKDD'02: Proceedings of the eighth ACM SIGKDD international
9. Yunbo CAO, Jun XU, Tie-Yan LIU, Hang LI, Yalou HUANG, Hsiao-Wuen HON, Adapting Ranking SVM to Document Retrieval, *SIGIR'06,* August 6-11,2006, Seattle, Washington, USA. pp.186∼193