A Probabilistic Ranking Approach for Tag Recommendation

Zhen Liao[1], Maoqiang Xie[2], Hao Cao[2], Yalou Huang[2]

[1] College of Information Technology Science, Nankai University, Tianjin, China
{liaozhen, caohao}@mail.nankai.edu.cn
[2] College of Software, Nankai University, Tianjin, China
{xiemq, huangyl}@nankai.edu.cn

Abstract. Social Tagging is a typical Web 2.0 application for users to share knowledge and organize the massive web resources. Choosing appropriate words as tags might be time consuming for users, thus a tag recommendation system is needed for accelerating this procedure. In this paper we formulate tag recommendation as a probabilistic ranking process, especially we propose a hybrid probabilistic approach which combines language model and statistical machine translation model. Experimental results validate the effectiveness of our method.

1 Introduction

Folksonomy is a way to categorize Web resources via utilizing the "wisdom" of web users, nowadays it is existing in many web applications such as Delicious[3], Filckr[4], Bibsonomy[5]. One user could create and share her knowledge during the tagging on resources that are interesting to her. Web resources come in many forms, for example, one resource could be a Web pages, a published paper, or a book. To tag a resource with appropriate words is not so easy and might cost lots of time. Thus a tag recommendation system is needed for easing the time-consuming step. Typically a recommendation system would suggest 5 or 10 tags to the user for a given resource. Those suggested tags would help one user to think about eligible words and to realize the interesting aspects concerned by others. To solve the problems, ECML PKDD holds the second round discovery challenge[6] of tag recommendation. This paper presents a probabilistic ranking approach submitted to the challenge.

Given a resource, users choose tags by different aspects of the resource and their specific interests. To pick up a tag from the entire tag set and assign it to the resource could be formulated as following process: given a resource and a user, ranking the tags by their *relevance* to the resource and user. Here relevance denotes the 'value' of how likely the user would label this tag on this resource.

[3] http://del.icio.us
[4] http://www.flickr.com/
[5] http://www.bibsonomy.org/
[6] http://www.kde.cs.uni-kassel.de/ws/dc09

We suppose a tag recommendation system works best while recommending tags are sorted by the relevance and then suggested to the user.

In this paper, the datasets provided by Bibsonomy is a set of *post*. Each *post* denotes a triple {user, resource, a set of tag}. A resource type could be *bookmark* or *bibtex*, where bookmark is Web page and bibtex is publication. Both bookmark and bibtex resources contain many fields: URL, description, etc. The textural information in the fields could be merged as a pseudo document.

A natural way of choosing tags is to select words from the pseudo document of given resource. A TF-like maximum likelihood method could reach the goal. The important problem is that maximum likelihood model could not generate tags which are meaningful but not existing in the document. To incorporate previously popular tags and tags preferred by a user, a tag recommendation model could be formulate into language model smoothed via Jelinek-Mercer method as described in Section 3.2. However, the language modeling approach could not learn the word-tag relateness which reflects how other users choose tags for those words in the document. Since the textural information existing in a post could be considered as a parallel corpus - {words in document, tags}, we propose to use the statistical machine translation approach to learn the translation probability from words to tags.

Finally, we propose a candidate set based tag recommendation algorithm which generates candidate tags from the textual fields of a resource using maximum likelihood and statistical machine translation model. The effectiveness of our approach is validated on the bookmark and bibtex tagging test datasets provided by Bibsonomy. While textural content of a bookmark resource is inadequate, we utilize the tags used within same Domain to extend the candidate set. We also found simple co-occurrence based translation probability estimation performs as good as IBM Model 1 [6] which uses the EM algorithm to learn the translation probability. An advantage of co-occurrence based approach is its convenience for handling with new training data, since training the model is just counting the co-occurrence of words and tags. However, EM-based approach needs to re-train translation model though iterations which might be time consuming for large scale dataset.

The rest of this paper is organized as follows. In Section 2 the related work is surveyed. In Section 3 our content based tag recommendation models are presented, and the recommendation algorithm is described in Section 4. In Section 5 we descrbe the data format and preprocessing step, and experimental results are reported in Section 6. Finally in Section 7 we conclude this paper and give out some possible future research issues.

2 Related Work

Most of existing tag recommendation approaches are based on the textual information of the resource and previous interests of users. Up to now, the information retrieval, data mining and natural language processing techniques have been used for solving the tag recommendation problem.

Heymann et al. [1] use one of the largest crawls from the social bookmarking system Delicious and presents studies of the factors which could impact the performance of tag prediction. The predictability of tags is measured by some method such as entropy based metric. The tag-based association rule is proposed to assist tag predictions. The method of learning the word-tag relateness via association rule needs to tune the confidence and support to find meaningful rules, but we transfer it into the translation probability which could get the converged solution without tuning.

Tatu et al [2] uses document and user models derived from the textual content associated with URLs and publications by social bookmarking tool users. The natural language processing techniques are used to extract the concept(Part of Speech, etc.) from the textual information. WordNet[7] are used to stem the concepts and link synonyms. The difference between our work and theirs is that they expand the concept via WordNet, but do not have the word-to-tag translation probability such as from 'eclipse' to 'java'.

Lipczak [3] focus on the folksomomies towards individual users, and proposed a three step tag recommendation system which conducts the Personmony based filtering using previously used tags of users after the extraction and retrieving of tags. The recommendation approach in [3] is similar with our work, but the scores of candidate tags are computed differently. They use the multiply strategy for different factors, but we conduct a weighted sums in which the weight could be set to prefer different components. Besides, we use the statistical machine translation approach to learn the word-tag relateness which is different from model proposed in [3].

Language modeling approach [4] has been applied in Information Retrieval with lots of smoothing strategies [5]. The statistical machine translation approaches [6] shows its theoretical soundness and effectiveness in translation, and Berger et al [7] and Xue et al [8] incorporate the statistical translation approaches into information retrieval and automatic question answering fields. The theoretical soundness and effectiveness make it stable to adopt the language modeling and statistical machine translation approach into tag recommendation. The statistical machine translation approach also naturally solve the problem of learning the word-tag relateness of sharing the common tagging knowledge among users.

3 Content Based Tag Recommendation Models

3.1 Problem Definition

In this paper, a tag set is denoted as $\mathbf{t} = \{t_i\}_{i=1}^{Q}$ where t_i is a single word or term and Q is the number of tags in $\mathbf{t}$.

The tag recommendation task is to suggest a tag set $\mathbf{t}$ for a user U_k while given a bookmark/publication resource R_j which might be a web page, a book or paper etc. The resource R_j contains several fields such as URL, title, description and we denote the resource content as a pseudo document D_j.

[7] http://wordnet.princeton.edu

Suppose the recommendation system is required to suggest N tags, it is to find N tags $\{t_i\}_{i=1}^{N}$ from the entire tag sets with the biggest probability $p(t_i|U_k, D_j)$.

For solving the task, a training set $\mathbf{S} = \{S^i\}_{i=1}^{K}$ is given, where S^i specifies a triple $\{\mathbf{t}^i, U^i, D^i\}$. The $\mathbf{t}^i$ is a tag set, $U^i \in \mathcal{U} = \{U_1, ..., U_M\}$ is a user and $D^i \in \mathcal{D} = \{D_1, ..., D_N\}$ is a resource . Then we can learn a tag recommendation model $\mathcal{M}$ from $\mathbf{S}$.

At the testing stage, a testing set $\mathbf{T} = \{T^j\}_{j=1}^{P}$ where $T^j = \{U^j, D^j\}$ is given. The model $\mathcal{M}$ is asked to suggest tag set $\mathbf{t}^j$ for each T^j. After that a groudtruth tag sets $G = \{\mathbf{g}^j\}_{j=1}^{P}$ is used to judge the recommendations $\{\mathbf{t}^j\}_{j=1}^{P}$, and the performance is get via some evaluation measures such as Precision, Recall and F-measure.

For a specific user U_k, she would have her preference in choosing a word t_i as a tag, and if we have this user's information in the training set $\mathbf{S}$, we can formulate this preference as $P(t_i|U_k) = \frac{c(t_i;U_k)}{|U_k|}$ where $c(t_i; U_k)$ is frequency of t_i be used by user U_k, and $|U_k|$ is total frequency of all tags used by U_k.

We define the tag generating probability a tag t_i for a given user and document tuple $\{U_k, D_j\}$ as:

$$P(t_i|D_j, U_k) = (1 - \beta)P(t_i|D_j) + \beta P(t_i|U_k) \tag{1}$$

Where β is a trade-off parameter between the resource content and user.

Following we will introduce language model and statistical machine translation approaches for estimating $P(t_i|D_j)$, and then we will combine them into our final model.

3.2 Language Modeling Approach

A natural and simple way to estimate $P(t_i|D_j)$ is to use the maximum likelihood approach as:

$$P_{ml}(t_i|D_j) = \frac{c(t_i; D_j)}{|D_j|} \tag{2}$$

Where $c(t_i; D_j)$ is occurrence of t_i in D_j, and $|D_j|$ is document length of D_j. The shortcoming of the maximum likelihood estimation is that it could not generate tag which does not exist in D_j, thus we introduce language model smoothed via Jelinek-Mercer method [5] as:

$$P_{lm}(t_i|D_j) = (1 - \lambda)P_{ml}(t_i|D_j) + \lambda P_{ml}(t_i|C) \tag{3}$$

Where λ is the smoothing parameter, and C corresponds to the entire corpus. Actually the smoothing term $P(t_i|C)$ could be formulated as the probability of the word t_i be used as a tag. We define $P(t_i|C)$ as $\frac{c(t_i)}{\#tags}$ where $\#tags$ is the total number of tags in the training set $\mathbf{S}$. The language modeling approach (3) could be considered as the incorporation of words in the document and previously popular tags of all users.

3.3 Statistical Machine Translation Approach

However, the language modeling approach has not considered word-tag relateness which would be important for tag recommendation. For solving the problem, we further introduce the Statistical Machine Translation(SMT) approach [6] [7] [8] for estimating the probability $P(t_i|D_j)$:

$$P_{smt}(t_i|D_j) = \frac{|D_j|}{|D_j|+1}P_{tr}(t_i|D_j) + \frac{1}{|D_j|+1}P(t_i|null) \tag{4}$$

Where $P(t_i|null)$ could be regarded as the background smoothing model $P(t_i|C)$, and a more detailed comparison them could be found in [8]. $P_{tr}(t_i|D_j)$ is the translation probability from D_j to ti as following:

$$P_{tr}(t_i|D_j) = \sum_{w \in D_j} P_{tr}(t_i|w)P_{ml}(w|D) \tag{5}$$

To learn the word-word transition probability $P_{tr}(t_i|w)$, the EM algorithm could be used. The detail of EM algorithm of learning the word-tag relateness $P(t_i|w)$ in Statistical Machine Translation(SMT) Model is described in [6]. In the training set $\mathbf{S} = \{S^j\}_{j=1}^K$, the parallel corpus of tag and document as $S^j = \{\mathbf{t}^j, D^j\}$ is utilized, and the EM step for learning $P(t_i|w)$ can be formulated as:

E-Step:

$$P_{tr}^1(t_i|w) = \delta_w^{-1} \sum_{j=1}^{K} c(t_i, w; \mathbf{t}^j, D^j) \tag{6}$$

M-Step:

$$c(t_i, w; \mathbf{t}^j, D^j) = \frac{P(t_i|w)}{P(t_i|w_1) + ... + P(t_i|w_o)} \#(t_i, \mathbf{t}^j) \#(w, D^j) \tag{7}$$

In Equation (6) $\delta_w^{-1} = \sum_{t_i} \sum_{j=1}^{K} c(t_i, w; \mathbf{t}^j, D^j)$ is the normalization factor. In Equation (7) $\{w_1, ..., w_o\}$ is words contained in D^j, $\#(t_i, \mathbf{t}^j)$ and $\#(w, D^j)$ is the number of t_i in $\mathbf{t}^j$ and number of w in D^j. The convergency of this EM algorithm is proved in [6].

In this paper, we also find that the co-occurrence based translation probability could be helpful in tag recommendation, and we denote it as:

$$P_{tr}^2(t_i|w) = \frac{\sum_{j=1}^{K} \#(t_i; \mathbf{t}^j) \cdot \#(w; D^j)}{\sum_{j=1}^{K} \#(w; \mathbf{t}^j, D^j)} \tag{8}$$

Where $\#(t_i; \mathbf{t}^j)$ denotes the number of tag t_i exists in $\mathbf{t}^j$ and the same to $\#(w; D^j)$. This model could be regarded as a simple approximation of the EM based translation model, and it is also effective. Note that the EM based translation probability is denoted as $P_{tr}^1(t_i|w)$ whereas the co-occurrence based translation probability is denoted as $P_{tr}^2(t_i|w)$ hereafter.

3.4 Final Model

Now we combine above methods together to get our final model:

$$\begin{aligned} P_{final}(t_i|D_j,U_k) =& \lambda P(t_i|C) + \beta P(t_i|U_k) \\ &+ \alpha P_{ml}(t_i|D_j) + \gamma \sum_w P_{tr}(t_i|w) P_{ml}(w|D) \end{aligned} \tag{9}$$

Where $\lambda + \beta + \alpha + \gamma = 1$ and P_{tr} could be P^1_{tr} or P^2_{tr}. Tuning these four parameters is not easy, and thus we split both Cleaned Dump and Post Core dataset into a training set and a validation set respectively, train the model on the training set and set parameters empirically several times for choosing one with better performance on the validation set. We do not illustrate the detail due to space restriction, and in the experiments we found the performance is relatively well while $\lambda = 0.15$, $\beta = 0.1$, $\alpha = 0.05$, $\gamma = 0.7$. We use these parameters with Cleaned Dump dataset as our final training set for the challenge.

4 Candidate Set based Tag Recommendation Algorithm

Since the task of tag recommendation is to suggest tags for given document and user, it is different from the task of Information Retrieval [7] or Question Answering [8] where the query/question is given for finding the relevant documents/answers.

Given a document D_j and user U_k, we firstly find a recommendation tag candidate set CS from the words in D_j, and we also add the top L related words by $P_{tr}(t|w)$ for every word w in D^j. Then we compute the $P(t_i|D_j,U_k)$ for each tag $t_i \in CS$. Finally we sort the tags descending according to $P(t_i|D_j,U_k)$, and return the top N tags as required by the application system. The L is set to be 20 and N is set to 5 in the experiments. In summary, we get this algorithm in Table 1.

5 Data Preparing and Preprocessing

The dataset we used is download from ECML PKDD Discovery Challenge 2009[8] which is provided by BibSonomy[9]. There are two datasets: **Cleaned Dump** and **Post Core**. The Cleaned Dump contains all public bookmarks and publication posts of BibSonomy until (but not including) 2009-01-01. The Post Core is a subset of the Cleaned Dump, it removes all users, tags, and resources which appear in only one post from Cleaned Dump. Brief statistics of Cleaned Dump and Post Core could be found in Table 2. One tag assignment means one user choose a tag for a resource, and thus one posts could have several tag assignments. The number of posts are shown for bookmark, bibtex, and entire set. The bookmark and bibtex are seperated by '/', and the entire set are illustrated after ':'.

[8] http://www.kde.cs.uni-kassel.de/ws/dc09
[9] http://www.bibsonomy.org/

Table 1. Candidate Set based Tag Recommendation Algorithm

Input: testing sample: $T^j = \{D^j, U^j\}$, threshold N and L
Output: top N tags $\mathbf{t} = \{t_1, ..., t_N\}$

1. candidate set $CS \leftarrow \emptyset$
2. for w in D^j
3. add w into CS
4. add top L tags t into CS according to $P(t|w)$
5. end for
6. for each word $t_k \in C$
7. compute $P(t_k|D^j)$ using (9)
8. end for
9. sort $t_k \in CS$ with $P(t_k|D^j)$ in descending order
10. return top N tags in C as $\mathbf{t}$

Table 2. Statistics of Cleaned Dump & Post Core datasets

	tag assignments	number of posts	number of users
Cleaned Dump	1,401,104	263, 004 / 158, 924 : 421, 928	3, 617
Post Core	253,615	41,268 / 22,852 : 64, 120	1, 185

There are three tables *tas*, *bookmark*, and *bibtex* in the dataset. The fields of these tables are list in Table 3. For bookmark resource the field 'content_type' is 1 and that of bibtex resource is 2. The fields in bold are used to generate the pseudo document D_j and the tags $\mathbf{t}^j$ in the training process.

Table 3. Fields of Three Dataset Tables

table	fields
tas	user, **tag**, content_type, content_id, date
bookmark	content_id, url_hash, URL, **description**, **extended description**, date
bibtex	content_id, journal, chapter, edition, month, day, **booktitle**, howPublished, institution, organization, publisher, address, school, series, bibtexKey, url, type, **description**, annote, note, pages, bKey, number, crossref, misc, bibtexAbstract, simhash0, simhash1, simhash2, entrytype, **title**, author, editor, year

We firstly remove the stop words in the bookmark and bibtex table since they are seldom used as tags and usually meaningless. The stop word list are download from Lextek[10]. Note that we do not remove stop words in the tas file, and the top 5 stop words exist in Post Core and their frequency could be found in Table 4. There are totally 19, 647 and 2, 513 stop word tag assignments in Cleaned Dump and Post Core, corresponds to 1.39% and 0.99% respectively.

[10] http://www.lextek.com/manuals/onix/stopwords1.html

In contrast, the total frequency of stop words in pseudo documents of Cleaned Dump and Post Core are over 588, 907 and 61, 113, which suggest not to consider stop words as tags in most cases.

Table 4. Top 5 stop words in tags of Cleaned Dump & Post Core

dataset	top 5 stop words and their frequency in tags
Cleaned Dump	all:3105 of:1414 and:1227 best:1124 three:1081 c:806
Post Core	all:655 open:211 c:165 best:152 work:77

In Table 5 we list out the top 10 tags in Cleaned Dump and Post Core. We could see later that the co-occurrence based translation model are likely to generate words which appear more times.

Table 5. Top 10 Tags and their Frequency

Cleaned Dump	bookmarks:52795 → zzztosort:11839 → video:10788→ software:10171 → programming:9491 → indexforum:9183 → web20:8777 → books:7934 → media:7149 → tools:6903
Post Core	web20:4474 → software:3867 → juergen:3092 → tools:3058 → web:2930 → tagging:2196 → semanticweb:2055 → folksonomy:1944 → search:1896 → bookmarks:1840

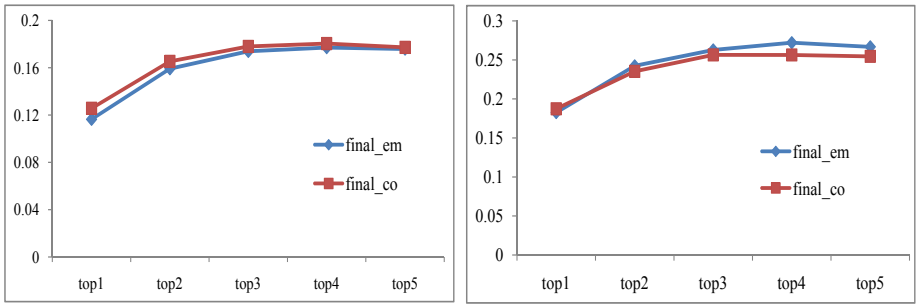
6 Experimental Result

6.1 Tagging Performance

The evaluation measure in following experiments are widely used Precision, Recall, and F1-measure. The testing datasets are released by ECML-PKDD challenge in tasks. There are 2 tasks: task 1 and task 2, where task 1 is for content based tag recommendation, and task 2 is for graph based tag recommendation[11]. In task 1 the user, resource of a post might not exist before, so the content information of the resource would be critical for tag recommendation. In task 2 user, resource, and tags of each post in the test data are all contained in the Post Core dataset, thus it intends for methods relying on the graph structure of the training data only.

We use the whole Cleaned Dump dataset as the training set to train the model and test the performance of our model on both tasks. For choosing the parameters, we set $\alpha = 0.15, \lambda = 0.05, \beta = 0.1, \gamma = 0.7$ as mentioned before in Section 3.4. The results are shown in Figure 1. The final_em denotes final model with P^1_{tr}(EM-based), and final_co denotes final model with P^2_{tr}(Co-occurrence based). The x-axis is the top position and y-axis is the f-measure.

[11] http://www.kde.cs.uni-kassel.de/ws/dc09



(a) On Task 1 Testing Set (b) On Task 2 Testing Set

Fig. 1. Performance of Selected Models

The results indicates that although P^2_{tr}(Co-occurrence) is more simpler, it is comparable to P^1_{tr}. In our previous experiment, we also found sometimes the textual information from the bookmark resource are not adequate enough to generate some tags in the post and it needs to be expanded. Instead of using extrinsic resource such as WordNet, we aggregate the tags in the same web site domain for bookmark resource, and use them to expand the recommendations. The reason we don't expand the term in bibtex is because resources in bibtex are publication and the web site provide less information about tags. Also, trying other tag expansion methods would be our future work. We formulate this expansion as $P(t_i|Site)$, and the recommendation model for bookmark would become:

$$\begin{aligned} P_{final\_ex}(t_i|D_j,U_k) =& \lambda P(t_i|C) + \beta P(t_i|U_k) + \alpha P_{ml}(t_i|D_j) \\ &+ \gamma \sum_w P_{tr}(t_i|w) P_{ml}(w|D) + \theta P(t_i|Site) \end{aligned} \tag{10}$$

For illustrate the expansions of different domains, we sample some domains and their top used tags with the probability in Table 6.

Table 6. Sample Domains with Top 5 used tags

domain	tags and their previously used probability
www.apple.com	apple:0.17 mac:0.13 software:0.09 osx:0.07 bookmarks:0.07
answers.yahoo.com	knowledge:0.14 yahoo:0.14 web20:0.07 all:0.07 answer:0.07
ant.apache.org	java:0.19 ant:0.17 programming:0.07 apache:0.07 tool:0.07
picasa.google.com	google:0.21 image:0.14 download:0.14 linux:0.14 picasa:0.14
research.microsoft.com	microsoft:0.10 research:0.09 people:0.04 social:0.04 award:0.03
www.research.ibm.com	ibm:0.11 datamining:0.07 software:0.04 machinelearning:0.04 journal:0.04

After the tag expansion via the URL domain, the candidates set CS for the recommendation will have top used tags in the same domain of D_j. The performance of (10) with the expansions on the testing set are shown in Table 7 and 8. The performance are shown for only bookmark, only bibtex, and on entire set. The bookmark and bibtex are seperated by '/', and the entire set are illustrated after ':'. We choose the co-occurrence based model P^2_{tr} in the competition, and actually the performance in terms of F-measure at 5 is also good when using EM-based model P^1_{tr}. The F-measure of EM-based model with the same parameters as Table 7 for task 1 and task 2 are shown in Table 9. We can find that the P^2_{tr} and P^1_{tr} are comparable once again, on F-measure at 1, the Co-occurrence based model are better, but on F-measure at 5, the EM-based model are better.

Table 7. Performance for Task 1 ($\alpha = 0.15, \lambda = 0.05, \beta = 0.05, \gamma = 0.5, \theta = 0.25$ for bookmark, $\alpha = 0.15, \lambda = 0.05, \beta = 0.1, \gamma = 0.7$ for bibtex with P^2_{tr})

TOP N	Recall	Precision	F-Measure
1	0.0702 / 0.0975 : 0.0809	0.2232 / 0.3056 : 0.2556	0.1067 / 0.1477 : 0.1229
2	0.1116 / 0.1584 : 0.1300	0.1905 / 0.2584 : 0.2172	0.1406 / 0.1961 : 0.1624
3	0.1412 / 0.2011 : 0.1648	0.1664 / 0.2251 : 0.1895	0.1525 / 0.2120 : 0.1760
4	0.1636 / 0.2318 : 0.1904	0.1489 / 0.2000 : 0.1690	0.1556 / 0.2143 : 0.1787
5	0.1810 / 0.2563 : 0.2106	0.1339 / 0.1802 : 0.1521	0.1536 / 0.2111 : 0.1762

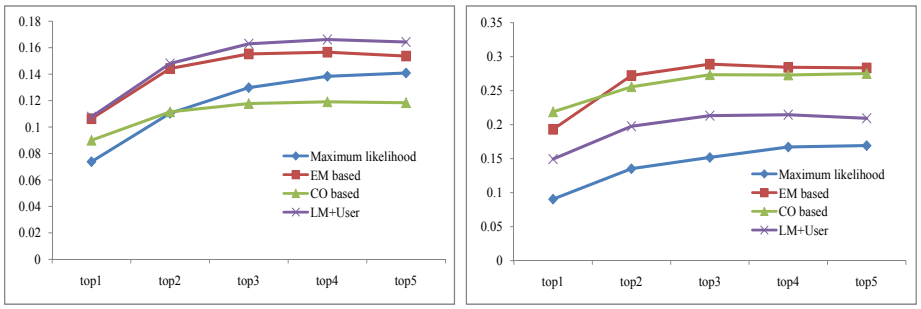
Table 8. Performance on Task 2 data($\alpha = 0.15, \lambda = 0.05, \beta = 0.05, \gamma = 0.5, \theta = 0.25$ for bookmark, $\alpha = 0.15, \lambda = 0.05, \beta = 0.1, \gamma = 0.7$ for bibtex with P^2_{tr})

TOP N	Recall	Precision	F-Measure
1	0.1399 / 0.1215 : 0.1297	0.4063 / 0.3666 : 0.3843	0.2073 / 0.1823 : 0.1938
2	0.2136 / 0.1919 : 0.2016	0.3444 / 0.3086 : 0.3246	0.2625 / 0.2365 : 0.2485
3	0.2887 / 0.2379 : 0.2605	0.3093 / 0.2676 : 0.2862	0.2977 / 0.2517 : 0.2726
4	0.3212 / 0.2848 : 0.3010	0.2630 / 0.2454 : 0.2532	0.2883 / 0.2636 : 0.2749
5	0.3532 / 0.3220 : 0.3359	0.2346 / 0.2237 : 0.2285	0.2812 / 0.2639 : 0.2718

Next we conduct the experiment on each component of our final model (9), the document maximum likelihood method, language model('LM + User Model'), the EM-based translation model $P^1_{tr}(t_i|w)$, and co-occurrence based translation model $P^2_{tr}(t_i|w)$ are chosen. In the 'LM + User Model' we set the parameters $\alpha = 0.5, \lambda = 0.3, \beta = 0.2, \gamma = 0$. It could be considered as the language model which incorporates the maximum likelihood, the previously tag probability in the whole corpus, and the user's preference model. The performance on both testing datasets of task 1 and task 2 are illustrated in Figure 2. The x-axis is the top position from top1 to top5 and the y-axis is the value of F-Measure. We only list out the F1 measure because it reflects both precision and recall.

Table 9. Performance of ($\alpha = 0.15, \lambda = 0.05, \beta = 0.05, \gamma = 0.5, \theta = 0.25$ for bookmark, $\alpha = 0.15, \lambda = 0.05, \beta = 0.1, \gamma = 0.7$ for bibtex with P^1_{tr})

TOP N	task 1 F-Measure	task 2 F-measure
1	0.1167	0.1909
2	0.1593	0.2548
3	0.1745	0.2790
4	0.1778	0.2866
5	0.1770	0.2833



(a) On Task 1 Testing Set (b) On Task 2 Testing Set

Fig. 2. Performance of Selected Models

From the experimental results we can see the translation based models are better than maximum likelihood method and 'LM + User Model' in task 2. The co-occurrence based model are worst in task 1, and the EM-based model is better than co-occurrence based model on both task. We analyze the results of co-occurrence based model on task 1 and find many recommendations are common used tags, because the co-occurrence based model would prefer to generate those tags occurred more times before. This suggest that if the resource/users have been seen before, thus the co-occurrence based model would perform well, if not, then it is better to choose EM based model. The 'LM + User Model' perform best on task 1, but the performance is still lower than that in Table 7, and also, 'LM + User Model' performs worse than translation models on task 2.

For comparison between EM-based and co-occurrence based model, we pick out several words w with their top translating words t_i in both $P^1_{tr}(t_i|w)$(EM-based) and $P^2_{tr}(t_i|w)$(Co-occurrence based). The sampling words could be found in Table 10. We could find that in EM-based translation model, the words are most likely to translate into itself. It indicates that we could consider the EM-based translation model as the combination of the maximum likelihood which only generates the word it self and the co-occurrence based translation model which has higher probability to generate other words as tags. The co-occurrence model are likely to generate those popular tags in the corpus, such as 'tools', 'software', 'social'.

Table 10. Sampled Words with their top tags t_i: $P^1_{tr}(t_i|w)$(EM); P^2_{tr}(CO)

w	model	Top tags t_i with highest probability $P_{tr}(t_i\|w)$
web	EM	web:0.36 web20:0.26 semanticweb:0.12 semantic:0.01970 ajax:0.02
	CO	web20:0.05 semanticweb:0.04 web:0.04 semantic:0.02 tools:0.01
wiki	EM	wiki:0.85 web20:0.01 semantic:0.01 wikipedia:0.01 collaboration:0.01
	CO	wiki:0.15 semantic:0.03 semanticweb:0.03 web20:0.02 software:0.02
dynamics	EM	dynamics:0.18 loreto:0.06 tagging:0.05 rmpcfl:0.04 analysis:0.04
	CO	tagging:0.07 dynamics:0.04 folksonomy:0.03 juergen:0.03 social:0.02
eclipse	EM	eclipse:0.55 java:0.23 development:0.05 ide:0.03 plugin:0.02
	CO	eclipse:0.18 java:0.13 plugin:0.06 develop:0.04 tools:0.04
yahoo	EM	yahoo:0.52 search:0.09 news:0.04 bookmarks:0.03 email:0.02
	CO	yahoo:0.09 search:0.04 web20:0.02 web:0.02 news:0.02

7 Conclusion and Future Work

In this paper we propose a probabilistic ranking approach for tag recommendation. The textual information from the resources and the parallel textual corpus from previously posts are used to learn the language and statistical translation model. Our hybrid probabilistic approach incorporates both the content based textural model and graph structure existing in posts for sharing the common tagging knowledge among users.

As our future work, we intent to study how to choose parameters via machine learning approaches to avoid heuristic setting. Further more, increasing the extra information of the resources, for example, using the citations(references) of a publication to augment the information of bookmark resource; using other tag expansion techniques; conducting the natural language understanding of the tag concept as well as studying the evaluation measures for tag recommendation are all possible future research work.

Acknowledgement

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. The authors thank Chin-Yew Lin at Microsoft Research Asia for his valuable comments to this paper. Thanks also to Jie Liu, Yang Wang and Min Lu for their helpful discussions and suggestions.

References

1. Heymann, P. and Ramage, D. and Garcia-Molina, H. Social Tag Prediction. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2008),* pages 531-538.
2. Tatu, M., Srikanth, M. and D'Silva, T. RSDC'08: Tag Recommendations using Bookmark Content. In *Proceedings of ECML PKDD Discovery Challenge 2008 (RSDC 2008),* pages 96-107.

3. Lipczak, M. Tag Recommendation for Folksonomies Oriented towards Individual Users. In *Proceedings of ECML PKDD Discovery Challenge (RSDC 2008),* pages 84-95.
4. Ponte, J. M. and Croft, W.-B. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1998),* pages 275-281.
5. Zhai, C.-X. and Lafferty, J. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transaction of Information System 2004,* pages 179-214.
6. Brown, P.-F., Pietra, V. J. D., Pietra, S. A. D. and Mercer, R.-L. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Journal of Computational Linguist 1993,* pages 263-311.
7. Berger, A. and Lafferty, J. Information Retrieval as Statistical Translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1999),* pages 222-229.
8. Xue, X., Jeon, J. and Croft., W.-B. Retrieval Models for Question and Answer Archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2008),* pages 475-482.