



Estimating translation probabilities for social tag suggestion



Xinxiong Chen*, Zhiyuan Liu, Maosong Sun

Department of Computer Science and Technology, State Key Lab on Intelligent Technology and Systems, National Lab for Information Science and Technology, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:

Available online 12 October 2014

Keywords:

Natural language processing
Tag suggestion
Translation model
Word alignment model
Pointwise mutual information

ABSTRACT

The task of social tag suggestion is to recommend tags automatically for a user when he or she wants to annotate an online resource. In this study, we focus on how to make use of the text description of a resource to suggest tags. It is intuitive to select significant words from the text description of a source as the suggested tags. However, since users can arbitrarily annotate any tags to a resource, tag suggestion suffers from the *vocabulary gap* issue – the appropriate tags of a resource may be statistically insignificant or even do not appear in the corresponding description. In order to solve the *vocabulary gap* issue, in this paper we present a new perspective on social tag suggestion. By considering both a description and tags as summaries of a given resource composed in two languages, tag suggestion can be regarded as a translation from description to tags. We propose two methods to estimate the translation probabilities between words in descriptions and tags. Based on the translation probabilities between words and tags estimated for a large collection of description-tags pairs, we can suggest tags according to the words in a resource description. Experiments on real-world datasets indicate that our methods outperform other methods in precision, recall and F-measure. Moreover, our methods are relatively simple and efficient, which makes them practical for Web applications.

© 2014 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

1. Introduction

In Web 2.0, Web users often use tags to collect and share online resources such as Web pages, photos, videos, movies and books. As an example, we consider a social tagging system for books. Table 1 presents a book entry annotated with several tags by multiple users.¹ On the top of Table 1 we list the title and a short introduction of the book “*The Count of Monte Cristo*”. The bottom of Table 1 shows the annotated tags, each of which is followed by a number in brackets, which is the total number of users who used the tag to annotate the book. As the tags of online resources are annotated collaboratively by multiple users, we also refer to these tags as *social tags*. For a resource, we refer to the additional information, such as the title and the introduction of a book, as *description*, and the user-annotated social tags as *annotation*.

The task of social tag suggestion is to automatically recommend tags for a user when he or she wants to annotate a resource. Social

tag suggestion, as a crucial component for social tagging systems, can help users annotate resources. Moreover, social tag suggestion is usually considered as an equivalent problem to modeling social tagging behaviors, which is playing an increasingly important role in social computing and information retrieval.

Most online resources have descriptions, usually containing abundant information about resources (Liu, Chen, & Sun, 2011). For example, on a book review website, each book entry contains a title, the author(s) and an introduction of the book. Thus, a number of researchers (Liu et al., 2011; Katakis, Tsoumakas, & Vlahavas, 2008; Mishne, 2006; Xu, Fu, Mao, & Su, 2006) propose to automatically suggest tags based on resource descriptions, which is collectively known as the *content-based* approach (Xu et al., 2006). In this study, we focus on how to make use of the text description of a resource to suggest tags. Note that besides descriptions, online resources may also have multimedia data (e.g., images, videos and audio files) and a survey of multimedia tagging can be found in Wang, Ni, Hua, and Chua (2012).

One may think to suggest tags by selecting important words from descriptions. This approach is far from sufficient because descriptions and annotations use diverse vocabularies, which is typically referred to as the *vocabulary gap* problem (Liu et al., 2011). The *vocabulary gap* is usually reflected in two primary issues:

* Corresponding author at: Room 4-506, FIT building, Tsinghua University, Beijing 100084, China. Tel.: +86 138 1035 7951.

E-mail addresses: cxx.thu@gmail.com (X. Chen), liuzy@tsinghua.edu.cn (Z. Liu), sms@tsinghua.edu.cn (M. Sun).

¹ The original record was obtained from the book review website Douban (www.douban.com) in Chinese. Here we translate it into English for comprehension.

Table 1

An example of social tagging. The number in the bracket after each tag is the total count of users that annotated the book with the tag.

Description	
Title:	The Count of Monte Cristo
Intro:	<i>The Count of Monte Cristo</i> is one of the most popular fiction by Alexandre Dumas. The writing of the work was completed in 1844. ...
Annotation	
Dumas (2748), Count of Monte Cristo (2716), foreign literature (1813), novel (1345), France (1096), classic (1062), revenge (913), famous book (759), ...	

1. A portion of tags in the annotation do appear in the corresponding description, but they may not be statistically significant.
2. A portion of tags may even not appear in the description.

Taking the book entry in Table 1 as example, the tag “classic” had been annotated by 1062 users but it did not appear in the description; another appropriate tag “famous book” also did not appear in the description.

Many approaches have been proposed to reduce the *vocabulary gap* and find the semantic correspondence between descriptions and annotations. Several researchers regard social tag suggestion as a classification problem by considering each tag as a category label (Fujimura, Fujimura, & Okuda, 2008; Heymann, Ramage, & Garcia-Molina, 2008; Katakis et al., 2008; Lee & Chun, 2007; Mishne, 2006; Ohkura, Kiyota, & Nakagawa, 2006). Various classifiers such as Naive Bayes, *k*NN and SVM have been explored and words are used as features. Some researchers propose to use the topic information between words and tags to suggest tags (Iwata, Yamada, & Ueda, 2009; Si, Liu, & Sun, 2010).

In this paper, we propose a new perspective on social tag suggestion to solve the *vocabulary gap* problem. By regarding both the description and the annotation as parallel summaries of a resource, we want to build a translation model to estimate the translation probabilities between the words in descriptions and tags in annotations. The translation probabilities are able to capture the semantic relation between words and tags. After obtaining the translation probabilities, the tagging behavior associated with a resource can then be regarded as a word translation process:

1. A user reads the resource description and understands its substance according to the important words in the description.
2. Triggered by the important words in the description, the user translate these words into corresponding tags and annotate the resource with these tags.

In Fig. 1, we provide a simple example to demonstrate the basic idea of using word translation for tag suggestion. In this figure, some words in the first sentence of book description are translated to tags in the annotation. The translation is denoted with various arrows from words or phrases in the description to tags in the annotation. For example, the phrase *Count of Monte Cristo* in the description is translated to two tags, including *Dumas* and *Count of Monte Cristo*, and the word *fiction*s is translated to *novel*.

In this paper, we propose two methods to estimate translation probabilities between words in descriptions and tags in

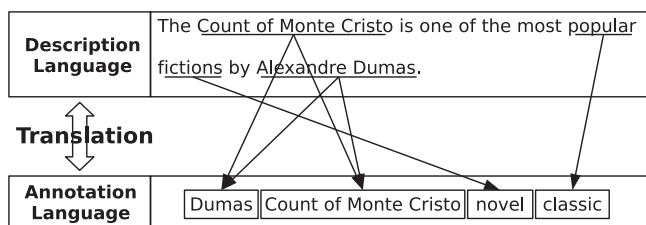


Fig. 1. An example of the word translation method for suggesting tags from a given description.

annotations. One method is the word alignment model (WAM) in statistical machine translation (SMT) and the other method is mutual information (MI) (Lin, 1998). It is straightforward to use WAM since it is the basic model in SMT to estimate the translation probabilities. For training, WAM requires a collection of parallel documents, where each document pair should have the comparable length. In this paper we propose a sampling method to prepare length-balanced description–annotation pairs for WAM. Moreover, we propose the second method, MI, to estimate the translation probabilities. Mutual information is a popular measure that can utilize co-occurrence information to measure semantic similarities between two words (Lin, 1998).

Our model can solve the *vocabulary gap* problem because the translation probabilities estimated by WAM and MI are able to capture the semantic relation between words and tags. Thus we can suggest tags that are not statistically significant or even not appear in the descriptions based on the translation probabilities. We hypothesize that our approach is better than the methods mentioned above and conduct experiments to investigate the performance of our model in the task of tag suggestion. Experiments on real-world datasets indicate that our method outperforms other methods in precision, recall and F-measure. Moreover, our method is relatively simple and efficient, as proven with the computational complexity, which makes it practical for Web applications.

The remainder of this paper is organized as follows: In Section 2 we briefly introduce some of the most commonly used methods for tag suggestion. Sections 3 and 4 introduce the details of our approach. Section 5 presents the experimental evaluation of our approach compared to other existing techniques. Finally Section 6 concludes the paper.

2. Related work

Many researchers have built social tag suggestion systems based on *collaborative filtering* (CF) (Herlocker, Konstan, Borchers, & Riedl, 1999; Herlocker, Konstan, Terveen, & Riedl, 2004). CF is a widely used technique in recommender systems (Resnick & Varian, 1997). The collaboration-based methods typically base their suggestions on the tagging history of the given resource and user, without considering resource description. Matrix Factorization (Rendle, Balby Marinho, Nanopoulos, & Schmidt-Thieme, 2009) and FolkRank (Jäschke, Marinho, Hotho, Schmidt-Thieme, & Stumme, 2008) are representative CF methods for social tag suggestion. Most of these methods suffer from the *cold-start problem* (Lam, Vu, Le, & Duong, 2008), i.e., they are not able to provide effective suggestions for resources that no one has annotated yet. The content-based approach for social tag suggestion ameliorates the *cold-start problem* of the *collaboration-based approach* by suggesting tags according to resource descriptions. Therefore, the content-based approach plays an important role in social tag suggestion, especially for new resources and new tagging systems without tagging history.

Several researchers regard social tag suggestion as a classification problem by considering each tag as a category label (Fujimura et al., 2008; Heymann et al., 2008; Katakis et al., 2008; Lee & Chun, 2007; Mishne, 2006; Ohkura et al., 2006). Various

classifiers such as Naive Bayes, k NN, SVM and neural networks have been explored to solve the social tag suggestion problem.

There are two issues emerging from the classification-based methods: (1) the annotations provided by users are noisy, and the classification-based methods cannot handle the issue well (Liu et al., 2011); (2) the training cost and classification cost of many classification-based methods are usually proportional to the number of classification labels (Si et al., 2010). Thus, these methods may be inefficient for a real-world social tagging system, where hundreds of thousands of unique tags should be considered as classification labels.

Inspired by the emerging popularity of latent topic models such as Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003), various methods have been proposed to model tags using generative latent topic models. One intuitive approach is assuming that both tags and words are generated from the same set of latent topics. By representing both tags and descriptions as the distributions of latent topics, this approach suggests tags according to their likelihood given the description (Krestel, Fankhauser, & Nejdl, 2009; Si & Sun, 2009). Bundschuh et al. (2009) proposed a joint latent topic model of users, words and tags. Iwata et al. (2009) proposed an LDA-based topic model, Content Relevance Model (CRM), which aims to identify the content-related tags for suggestion. Empirical experiments revealed that CRM outperformed both classification methods and Corr-LDA (Blei & Jordan, 2003), a generative topic model for contents and annotations.

Most latent topic models have to pre-specify the number of topics before training. We can either use cross-validation to determine the optimal number of topics or employ the infinite models for automatically adjusting the number of topics during training. Both of the solutions are usually computationally complicated. More importantly, topic-based methods suggest tags by measuring the topical relevance of tags and resource descriptions. The latent topics are at the concept level (Liu, Huang, Zheng, & Sun, 2010), which are usually too coarse-grained to precisely suggest fine-grained tags such as named entities, e.g., the tags “Dumas” and “Count of Monte Cristo” in Table 1. To remedy the problem, Si et al. (2010) proposed a generative model, Tag Allocation Model (TAM), which considers the words in descriptions as the possible topics to generate tags.

Our model is also a content-based approach so we compare some of the content-based approach mentioned above (k NN, Naive Bayes, CRM and TAM) to investigate the performance of our approach.

3. Learning translation probabilities

Given a resource description, the ranking score of a tag can be calculated from two probabilities: (1) the translation probabilities between words and tags, (2) the probabilities of a word given the description. In Section 3, we will present how to use two different methods, word alignment model and mutual information, to estimate the translation probabilities between words in description and tags in annotations. We will introduce how to calculate the probabilities of a word given the description and perform tag suggestion in Section 4.

First we give formal definitions to description and tags. In this paper, the description of a resource is the textual information of the resource, including the title and the short introduction. The description can be treated as a bag of words. Here we do not use stemming techniques or lemmatization techniques to preprocess the description. We just removed the stop-words in the description. Tags of a resource are labels with count information to describe the resource.

Before introducing the methods in details, we introduce the notation. A resource is denoted as $r \in R$, where R is the set of

resources. Each resource in the training set contains a description and an annotation containing a set of tags. The description d_r of resource r can be regarded as a bag of words $\mathbf{w}_r = \{(w_i, c_{w_i})\}_{i=1}^{N_r}$, where c_{w_i} is the count of word w_i , and N_r is the number of unique words in r . The annotated tags a_r of the resource r is represented as $\mathbf{t}_r = \{(t_i, c_{t_i})\}_{i=1}^{M_r}$, where c_{t_i} is the count of the tag t_i , and M_r is the number of unique tags for r .

3.1. Word alignment model (WAM)-based approach

WAM, as a traditional machine translation method, requires a parallel training dataset consisting of a number of aligned sentence pairs. We assume the description and the annotation of a resource are written in two distinct languages. Thus, we prepare our parallel training dataset by pairing descriptions and annotations. Accordingly, the WAM-based approach contains two steps. First, given a collection of annotated resources, we prepare description–annotation pairs for using the word alignment model. Second, given a collection of description–annotation pairs, we adopt IBM Model-1, a widely used word alignment model, to learn the translation probabilities between words in descriptions and tags in annotations. We will introduce the two steps separately.

3.1.1. Preparing description–annotation pairs for WAM

In a typical tag suggestion system, the length of a resource description is usually limited to hundreds of words. In addition, it is common for some popular resources to be annotated by multiple users with thousands of tags. For example, the tag *Dumas* is annotated by 2,748 users for the book in Table 1. In another extreme, a resource may be annotated with only several tags. We have to address the length-unbalance between a resource description and its corresponding annotation for two reasons. (1) When the number of annotated tags is large, it is impossible to list all annotated tags on the annotation side of a description–annotation pair. The performance of word alignment models will also suffer from the unbalanced length of aligned pairs in the parallel training data set (Och & Ney, 2003). (2) Moreover, the annotated tags may have different importance for the resource. It would be unfair to treat these tags without distinction.

In this study, we propose a sampling method to prepare length-balanced description–annotation pairs for word alignment. The basic idea is to sample a bag of tags from the annotation according to tag weights and make the generated bag of tags have comparable length with the words in description. For example, the length of the description in Table 1 is 54 and the number of unique tags is 21. If we list 54 words in one side and 21 tags in another side, we will get a sentence pair with unbalanced length. Thus we propose to sample a bag of tags with comparable length with the words, for example, 54 tags.

We consider two parameters when sampling tags. First, we have to select a **tag weighting type** for sampling. In this paper, we investigate two straightforward weighting types, including tag frequency (TF_t) within the annotation, which considers the local importance, and tag-frequency inverse-document-frequency ($TF\text{-}IDF_t$), which also considers global specification (IDF_w) besides TF_t . Given a resource r , TF_t and $TF\text{-}IDF_t$ of the tag t are defined as

$$TF_t = \frac{c_t}{\sum_t c_t}, \quad TF\text{-}IDF_t = \frac{c_t}{\sum_t c_t} \times \log \left(\frac{|R|}{|\{r \in R : c_t > 0\}|} \right) \quad (1)$$

where $|\{r \in R : c_t > 0\}|$ indicates the number of resources that have been annotated with the tag t .

Another parameter is the **length ratio** between the description and the sampled annotation. We denote the ratio as $\delta = \frac{|\mathbf{w}_r|}{|\mathbf{t}_r|}$, where $|\mathbf{w}_r|$ is the number of words in the description and $|\mathbf{t}_r|$ is the number of tags in the annotation. Still take the book in Table 1 for

example, if the length of the description is 54 and the length ratio is 10/5, then we will sample 27 tags from the annotations.

3.1.2. Learning translation probabilities for WAM

After preparing aligned description–annotation pairs for WAM, next we will choose an appropriate WAM model to obtain the translation probabilities between words in description and tags in annotations. Note that the annotated tags (\mathbf{t}_r) form a bag of labels with no position information, thus, we select IBM Model-1 (Brown, Pietra, Pietra, & Mercer, 1993) for training, which does not take word position information into account on both sides for each aligned pair.

Suppose the source language is the description, and the target language is the annotation. We use word alignment models to learn the translation probabilities between words in descriptions and labels in annotations. In IBM Model-1, the relationship of the source language $\mathbf{w} = w_1^j$ and the target language $\mathbf{t} = t_1^i$ is connected via a hidden variable describing an alignment mapping from source position j to target position a_j :

$$\Pr(w_1^j | t_1^i) = \sum_{a_j} \Pr(w_1^j, a_j | t_1^i). \quad (2)$$

The alignment a_j also contains empty-word alignments $a_j = 0$ which align source words to the empty word. IBM Model-1 can be trained using an Expectation–Maximization (EM) algorithm in an unsupervised fashion, and obtains the translation probabilities of two vocabularies, i.e., $\Pr(w|t)$, where t is a tag and w is a word.

IBM Model-1 only produces one-to-many alignments from source language to target language. The learned model is thus asymmetric, i.e., the model learned from description–annotation pairs is different from the model learned from annotation–description pairs. So we establish learned translation models in two directions: one regards description as the source language and annotations as the target language, and the other is in the reverse direction of the pairs. We denote the first model as \Pr_{d2a} and the latter as \Pr_{a2d} . We further define $\Pr(t|w)$ as the harmonic mean of the two models:

$$\Pr(t|w) \propto \left(\frac{\lambda}{\Pr_{d2a}(t|w)} + \frac{1-\lambda}{\Pr_{a2d}(t|w)} \right)^{-1}, \quad (3)$$

where λ is the harmonic factor for combining the two models. When $\lambda = 1$ or $\lambda = 0$, it simply uses model \Pr_{d2a} or \Pr_{a2d} correspondingly.

Finally we get the translation probabilities $\Pr(t|w)$ using the WAM model, which can be regarded as the semantic relatedness between words and tags.

3.2. Mutual information (MI)-based approach

From Section 3.1.1, we can see that WAM has to use a sampling technique to prepare description–annotation pairs. Unlike WAM, MI only needs the co-occurrence information between words in description and tags in annotations, which does not require sampling. We obtain translation probabilities using MI as follows.

First, for each pair of a word w in the descriptions and a tag t in the annotations, we compute their mutual information score. Informally, mutual information divides the probability of observing w and t together in the same resource by the probabilities of observing w and t independently. The mutual information between a word w and a tag t is calculated as follows:

$$I(w; t) = \sum_{X_w=0,1} \sum_{X_t=0,1} p(X_w, X_t) \cdot \log \frac{\Pr(X_w, X_t)}{\Pr(X_w)\Pr(X_t)} \quad (4)$$

where X_w and X_t are binary variables indicating whether w or t is present or absent, respectively. The estimation of the probabilities

$\Pr(X_w)$, $\Pr(X_t)$ and $\Pr(X_w, X_t)$ can be found in Karimzadehgan and Zhai (2010).

For a word w , we set a tag co-occurrence threshold γ and will remove the mutual information between word w and tag t if $c(X_w = 1, X_t = 1) \leq \gamma$, where $c(X_w = 1, X_t = 1)$ is the number of resources that contain both w and t . We set the co-occurrence threshold γ for two reasons. (1) We are usually less confident with the translation probabilities estimated using infrequent word-tag pairs, which are usually noisy and unimportant. With the threshold, we can filter out a lot of noisy information. (2) Moreover, we can largely reduce the computation cost of estimating mutual information scores. Of course, when $\gamma = 0$, we will not remove any word-tag pairs.

Thus, we normalize the mutual information score to obtain a translation probability $\Pr(t|w)$ between a word u and a tag t :

$$\Pr(t|w) = \frac{I(w; t)}{\sum_{t'} I(w; t')} \quad (5)$$

Intuitively, the probability is higher if the word u and the tag t are more likely to co-occur. For example, we get the probabilities $\Pr\{revenge|revenge\} = 0.127$ and $\Pr\{martialarts|revenge\} = 0.088$ from MI model, standing for the word “revenge” is more likely to co-occur with the tag “revenge” than the “martial arts”.

3.3. Emphasize self-translation probability

As a word may not always appear as a tag in annotation, the approaches described in Section 3.1 and Section 3.2 may underestimate the self-translation probabilities, i.e., it is possible that $\Pr(t \neq w|w) > \Pr(t = w|w)$. Here we propose to emphasize the self-translation probability for two reasons. (1) Under estimation of self-translation probabilities may lead to a situation where a proper tag t that also appears frequently in the description may receive a less recommendation score (i.e., $\Pr(t|w = t)$) compared to other tags t' (i.e., $\Pr(t'|w = t)$). (2) In some real tag suggest systems, the tags that appear in the resource description are more likely to be selected by users for annotation. Hence, we introduce a parameter α to emphasize self-translation probabilities. This idea can be applied to adjust the translation probabilities from any translation models.

$$\Pr(t|w) = \begin{cases} \alpha + (1 - \alpha)\Pr(t = w|w) & t = w \\ (1 - \alpha)\Pr(t|w) & t \neq w \end{cases} \quad (6)$$

When $\alpha = 1.0$, the method will suggest tags simply according to their importance scores in the description, whereas when $\alpha = 0$, it will not emphasize the tags that appear in the description and suggest tags only according to translation probabilities. For example, if $\alpha = 0.5$ and $\Pr\{revenge|revenge\} = 0.127$, then we will get a emphasized probability $\Pr\{revenge|revenge\} = 0.5635$.

4. Suggesting tags with translation probabilities

After estimating translation probabilities between words and tags $\Pr(t|w)$ in Section 3, we will show how to suggest tags in this section.

Given a resource description d_r , our model for tag suggestion is a 3-step process:

1. Measure the importance score $\Pr(w|d_r)$ of each word w in description d_r .
2. Compute the ranking score of tag t by

$$\Pr(t|d_r = \mathbf{w}_r) = \sum_{w \in \mathbf{w}_r} \Pr(t|w)\Pr(w|d_r) \quad (7)$$

3. According to $\Pr(t|d_r)$, suggest top-ranked tags to users.

Since we get $\Pr(t|w)$ in Section 3, we focus on how to measure $\Pr(w|d_r)$ in this section. Here we investigate three methods to compute the importance score of a word in a resource description: TF-IDF_w, TextRank and their product. TF-IDF_w and TextRank are the two most widely adopted methods to weight words in a document.

Similar to TF-IDF_t mentioned in Section 3.1.1, TF-IDF_w (Salton & Buckley, 1988) considers both the local importance (TF_w) and global specification (IDF_w). TextRank (Mihalcea & Tarau, 2004) is proposed to compute term importance using graph-based algorithms such as PageRank (Page, Brin, Motwani, & Winograd, 1998), which considers the semantic relations between terms as a term graph of the given resource description. We also use the product of TF-IDF_w and TextRank to weight terms, which potentially takes both global information and term relations into account.

Finally we get the ranking of tags according to Eq. (7). Because WAM and MI can estimate the translation probabilities between words and tags, which can be regarded as the semantic relatedness between words and tags, we hypothesize that our methods are better than the baseline methods mentioned in Section 2 (kNN, Naive Bayes, CRM and TAM). In next section we run experiments to validate this hypothesis.

5. Experiments

5.1. Datasets and evaluation metrics

Datasets In our experiments, we select two real-world datasets of social tagging systems that have diverse properties to evaluate our methods. In Table 2 we show the detailed statistical information of the two datasets.

The first dataset, denoted as BOOK, was obtained from a popular Chinese book review website, www.douban.com, which contains the descriptions of books and the tags collaboratively annotated by users.

The second dataset, denoted as BIBTEX, was obtained from an English online bibliography website, www.bibsonomy.org.² The dataset contains the descriptions for academic papers and the tags annotated by users.

As shown in Table 2, the average length of descriptions in the BIBTEX dataset is much shorter than the BOOK dataset. Moreover, the BIBTEX dataset does not provide how many times each label is used in a resource annotation.

5.1.1. Evaluation metrics

We use precision that measures exactness, recall that measures completeness and F-measure, which is the harmonic mean of precision and recall, to evaluate the performance of tag suggestion methods. For a resource, we denote the original tags (gold standard) as T_a , the suggested tags as T_s , and the correctly suggested tags as $T_s \cap T_a$. Precision, recall and F-measure are defined as follows:

$$p = \frac{|T_s \cap T_a|}{|T_s|}, \quad r = \frac{|T_s \cap T_a|}{|T_a|}, \quad F = \frac{2pr}{p+r}. \quad (8)$$

The final evaluation scores are computed by micro-averaging (i.e., averaging on resources of test set). We performed 5-fold cross-validation for each method for both datasets. In the experiments, the number of suggested tags M_r ranges from 1 to 10.

² The dataset can be obtained from <http://www.kde.cs.uni-kassel.de/bibsonomy/dumps>.

Table 2

Statistical information of two datasets. D , W , L , \bar{N}_d and \bar{N}_a are the number of resources, the vocabulary of descriptions, the vocabulary of tags, the average number of words in each description and the average number of tags in each resource, respectively.

Data	D	W	T	\bar{N}_d	\bar{N}_a
BOOK	70,000	174,748	46,150	211.6	3.5
BIBTEX	158,924	91,277	50,847	5.8	2.7

5.2. Comparing results

5.2.1. Baseline methods

We select four algorithms as the baselines for comparison: Naive Bayes (NB) (Manning, Raghavan, & Schtze, 2008), k nearest neighborhood (kNN) (Manning et al., 2008), Content Relevance (CRM) Model (Iwata et al., 2009) and Tag Allocation Model (TAM) (Si et al., 2010). Our methods are denoted as WAM and MI.

NB and kNN are two representative classification methods. NB is a simple generative model, which models the probability of each tag t given a description d as

$$\Pr(t|d) \propto \Pr(t) \prod_{w \in d} \Pr(w|t). \quad (9)$$

$\Pr(t)$ is estimated by the frequency of the documents annotated with the tag t . $\Pr(w|t)$ is estimated by the frequency of the word w in the resource descriptions annotated with the tag t . kNN is a widely used classification method for tag suggestion, which annotates tags to a resource according to the annotated tags of similar resources measured using vector space models (Manning et al., 2008).

CRM and TAM are selected to represent topic-based methods for tag suggestion. CRM is an LDA-based generative model. The number of latent topics K is the key parameter for CRM. In the experiments, we evaluated the performance of CRM with different K values, and here, we only present the best one obtained by setting $K = 1,024$. TAM is also a generative model that considers the words in descriptions as the topics to further generate tags for the resource. We set parameters for TAM as in Si et al. (2010).

We compare the complexity of these methods. We denote the number of training iterations in CRM, TAM and WAM as I , and the number of topics in CRM as K . For the training phase, the complexity of NB is $O(D\bar{N}_d\bar{N}_a)$; kNN is $O(1)$; TAM is $O(I\bar{N}_d\bar{N}_a)$; CRM is $O(IK\bar{N}_d\bar{N}_a)$; WAM is $O(I\bar{N}_d\bar{N}_a)$,³ and MI is $O(D\bar{N}_d\bar{N}_a)$. When suggesting for a given resource description with length N_d , the complexity of NB is $O(N_dT)$; kNN is $O(D\bar{N}_d\bar{N}_a)$; CRM is $O(IKN_dT)$; TAM is $O(I\bar{N}_dT)$; WAM is $O(N_dT)$, and MI is $O(N_dT)$. From the analysis, we can see that WAM and MI are relatively simple methods for both training and suggestion. This is especially valuable because WAM and MI also present good effectiveness for tag suggestion compared with other methods as we will shown later.

5.2.2. Parameter settings

For WAM, we use GIZA++ (Och & Ney, 2003)⁴ with the IBM Model-1 to determine translation probabilities using description-annotation pairs for WAM. The experimental results of WAM are obtained by setting parameters as follows: label weighting type as TF-IDF_t, length ratio $\delta = 1$, harmonic factor $\lambda = 0.5$ and the type of

³ In more detail, the training phase of WAM contains preparing parallel training dataset with $O(DN_d)$ and learning translation probabilities using word alignment models with $O(I\bar{N}_d\bar{N}_a)$, where I is the number of iterations for learning trigger probabilities, and \bar{N}_a is the average number of tags for each description after sampling.

⁴ GIZA++ is freely available from code.google.com/p/giza-pp. The toolkit is widely used for word alignment in SMT. In this paper, we use the default setting of parameters for training.

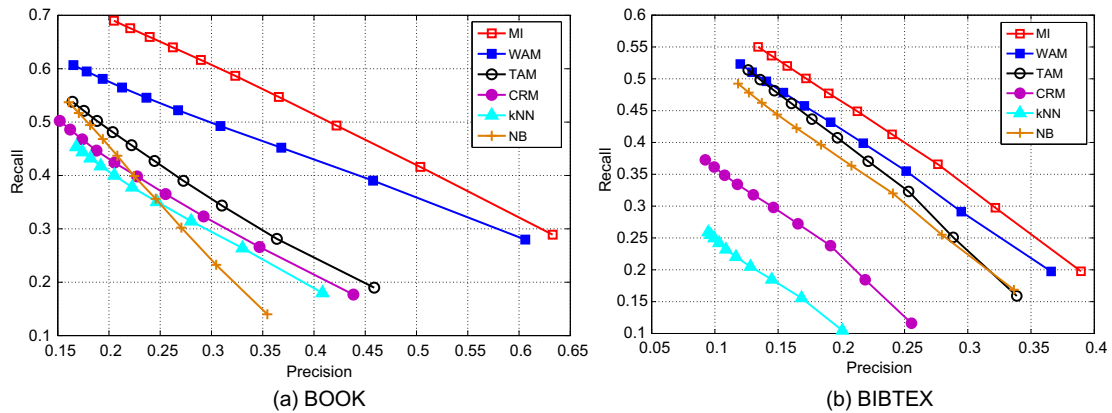


Fig. 2. Performance comparison between NB, kNN , CRM, TAM, WAM and MI for the two datasets.

word importance scores as $TF\text{-}IDF_w$. For MI, we set the tag co-occurrence threshold $\gamma = 0$ and the type of word importance scores as $TF\text{-}IDF_w$. The values are used as default values by maximizing the F-measure on a development set of 1000 instances from the website where BOOK dataset are obtained (not included in the BOOK dataset). The influence of parameters to WAM and MI can be found in Section 5.3.

5.2.3. Experiment results and analysis

In Fig. 2 we present the precision–recall curves of NB, kNN , CRM, TAM, WAM and MI for the two datasets. Each point of a precision–recall curve represents different numbers of tags from $M = 1$ (bottom right, with higher precision and lower recall) to $M = 10$ (upper left, with higher recall but lower precision). The closer the curve to the upper right is, the better the overall performance of the method is.

From Fig. 2, we observe the following:

1. The method based on MI consistently performs the best for both datasets. The method based on WAM achieves the second best performance for both datasets. These results indicate that our method is robust and effective for social tag suggestion.
2. The advantage of our method based on WAM is more significant on the BOOK dataset than the baseline method. The reason is that WAM has a good advantage in count information of tags compared with other baseline methods.
3. Although WAM has a good count information advantage for the BOOK dataset, MI is still better than WAM. The reason is that MI tries to translate a word to more tags, whereas the translation probabilities of a word in WAM always focus on only one or two tags. This leads to the result that the tags suggested by MI have a better coverage than WAM. We will present the translation probability table of words in the next subsection.
4. The average length of resource description is short in BIBTEX, which makes it difficult to determine the importance score of words, but even for the BIBTEX dataset with no count information of tags, our method still outperforms other methods.

To further demonstrate the performance of our word translation method and other baseline methods, in Table 3 we show the precision, recall and F-measure of NB, kNN , CRM, TAM, WAM and MI applied to the BOOK dataset when suggesting $M = 3$ labels.⁵ Here we also show the variance of F-measure. In fact, MI achieves the best performance when $M = 2$, where the F-measure of MI is

⁵ We elected to show this number because it is near the average number of labels for the BOOK dataset.

Table 3

Comparison of NB, kNN , CRM, TAM, WAM and MI results for the BOOK dataset when suggesting $M = 3$ tags. We have used t-test to ensure the differences between other results and the best result (in bold) in each column of the table are statistically significant at $p < 0.05$.

Method	Precision	Recall	F-measure
NB	0.271	0.302	0.247 ± 0.004
kNN	0.280	0.314	0.258 ± 0.002
CRM	0.292	0.323	0.266 ± 0.004
TAM	0.310	0.344	0.283 ± 0.001
WAM	0.368	0.452	0.355 ± 0.002
MI	0.422	0.493	0.397 ± 0.002

0.399, indicating an outperforming of both CRM ($F = 0.263$) and TAM ($F = 0.277$) by more than 10%.

5.2.4. An example

In Table 4, we show the top 10 tags suggested by NB, CRM, TAM, WAM and MI applied to the book in Table 1. The number in brackets after the name of each method is the count of correctly suggested labels. The correctly suggested tags are marked in bold face. We elected not to show the kNN results because the tags suggested by kNN are totally unrelated to the book due to the insufficient finding of nearest neighbors.

From Table 4, we observe that NB, CRM and TAM, as generative models, tend to suggest coarse-grained tags, such as “novel”, “literature”, “classic” and “France”, and fail in suggesting fine-grained tags such as “Alexandre Dumas”, “Count of Monte Cristo”, “revenge” and “suspense”. On the contrary, WAM and MI succeed in suggesting both the coarse-grained and the fine-grained tags related to the book.

To find out how our model can suggest these fine-grained tags, we list four important words (using $TF\text{-}IDF_w$ as weighting metric) of the description and their corresponding tags with the highest translation probabilities in Tables 5 and 6. The values in brackets are the probability of tag t given word w , $\Pr(t|w)$. For each word, we eliminated the tags with a probability less than 0.05. We can see that the translation probabilities can map the words in descriptions to their semantically corresponding tags in annotations. Take the word “Count of Monte Cristo” in Table 5 for example, besides the tag identical to itself, it has a high probability to the tag “Alexander Dumas”, which indicates the tag “Alexandre Dumas” is highly related to the word “Count of Monte Cristo”. In fact, the word “Count of Monte Cristo” appears in 19 books (12 of them are the different versions of “Count of Monte Cristo” and other are the novels written by “Alexander Dumas”) and 16 of them

Table 4

Top 10 tags suggested by NB, CRM, TAM, WAM and MI for the book in Table 1.

NB (+6): novel, foreign literature, literature, history , Japan, classic, France , philosophy, America, biography
CRM (+5): novel, foreign literature, literature , biography, philosophy, culture, France , British, comic, history
TAM (+5): novel , sociology, finance, foreign literature, France, literature , biography, France literature , comic, China
WAM (+7): novel, Alexandre Dumas, history, Count of Monte Cristo, foreign literature , biography, suspense , comic, America, France
MI (+7): Alexandre Dumas, novel, Count of Monte Cristo, foreign literature, France, revenge, French literature , Liang Yusheng, martial arts, Comedie Humaine

Table 5

Four important words (in bold face) in the book description in Table 1 and their corresponding tags with the highest translation probabilities in WAM.

Count of Monte Cristo: Count of Monte Cristo (0.728), Alexandre Dumas (0.270), ...
Alexandre Dumas: Alexandre Dumas (0.966), ...
Revenge: foreign literature (0.168), classic (0.130), martial arts (0.123), Alexandre Dumas (0.122), ...
France: France (0.99), ...

Table 6

Four important words (in bold face) in the book description in Table 1 and their corresponding tags with the highest translation probabilities in MI.

Count of Monte Cristo: Count of Monte Cristo (0.274), Alexandre Dumas (0.244), revenge (0.093), French literature (0.069), France (0.057), ...
Alexandre Dumas: Alexandre Dumas (0.352), France (0.105), French literature (0.069), Count of Monte Cristo (0.067), foreign literature (0.053), revenge (0.052), ...
Revenge: Liang Yusheng (0.154), revenge (0.127), martial arts (0.088), ...
France: France (0.309), France literature (0.069), ...

are labeled with the tag “Alexander Dumas”. This proves that our model can capture the semantic relation between words and tags.

Note that “Count of Monte Cristo” and “Alexander Dumas” correspond to the title and the author of the book in Table 1, they may be easily derived from other metadata of the book (although it is not the case of our dataset). So it is more interesting to see that our model can suggest the fine-grained tags like “revenge”. From Table 6, we can see that each of the words “Count of Monte Cristo”, “Alexander Dumas” and “revenge” has a probability to the tag “revenge”. The tag “revenge” is suggested jointly by combining the scores from these important words in the description. The ability to suggest a tag jointly enables our model to suggest the tags that are not statistically significant or even not appear in the descriptions. Thus our model can solve the *vocabulary gap* problem.

5.3. Parameter influences

5.3.1. Parameter influences for WAM

We explore the parameter influences on WAM for tag suggestion. The parameters include harmonic factor, length ratio, tag weighting types, and types of word translation power. When investigating one parameter, we set the other parameters to the values inducing the best performance as mentioned in Section 5.2. Finally, we also investigated the influence of training data size for classification performance. In the experiments we found that WAM reveals similar trends for both the BOOK dataset and the BIBTEX dataset. Thus, we only show the experimental results for the BOOK dataset for analysis.

5.3.1.1. Harmonic factor. In Fig. 3 we investigate the influence of harmonic factor via the curves of F-measure of WAM versus the number of suggested tags on the BOOK dataset when harmonic factor λ ranges from 0.0 to 1.0. As shown in Section 3.1.2, the harmonic factor λ controls the proportion between model Pr_{d2a} and Pr_{a2d} .

From Fig. 3, we observe that neither single model Pr_{d2a} ($\lambda = 1.0$) nor Pr_{a2d} ($\lambda = 0.0$) achieves the best performance. When the two models are combined by the harmonic mean, the performance is consistently better, especially when λ ranges from 0.2 to 0.6. This

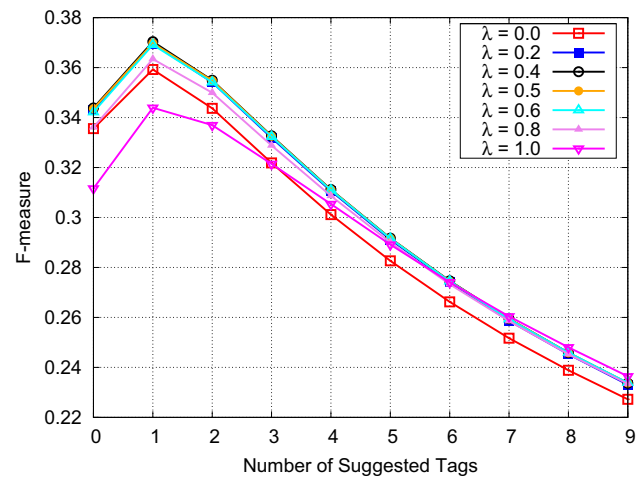


Fig. 3. F-measure of WAM versus the number of suggested tags for the BOOK dataset when harmonic factor λ ranges from 0.0 to 1.0.

is reasonable because IBM Model-1 constrains that only the term in the source language that can be aligned to multiple terms in target language, which makes the translation probability learned by a single model asymmetric.

5.3.1.2. Length ratio. Fig. 4 shows the influence of length ratios for WAM on the BOOK dataset. From the figure, we observe that the performance for tag suggestion is robust as the length ratio varies, except when the ratio breaks the default restriction of GIZA++ (i.e., $\delta = 10$).⁶

5.3.1.3. Tag weighting types. The influence of two weighting types, TF_t and TF-IDF_t , on tag suggestion when $M = 3$ on the BOOK dataset is shown in Table 7. TF-IDF_t tends to select the tags more specific to

⁶ GIZA++ restricts the values of length ratio within $[\frac{1}{9}, 9]$ by setting the parameter $\text{maxfertility} = 10$. From Fig. 4, we can see when $\delta = 10$, the performance becomes much worse as GIZA++ will cut off the sentences out of range.

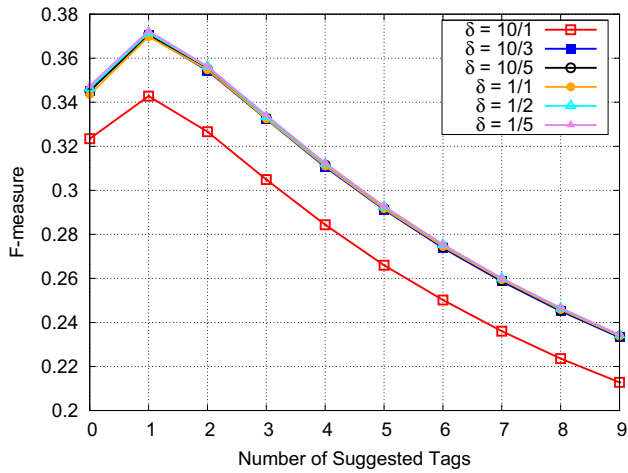


Fig. 4. F-measure of WAM versus the number of suggested tags for the BOOK dataset when length ratio δ ranges from 10/1 to 1/5.

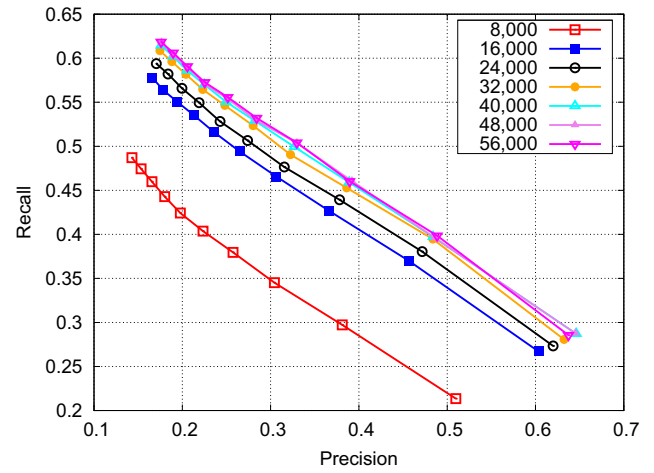


Fig. 5. Precision–recall curves of WAM when the training data size increases from 8000 to 56,000 on the BOOK dataset.

Table 7
Evaluation results for different tag weighting types for WAM when $M = 3$ on the BOOK dataset.

Weighting	Precision	Recall	F-measure
TF_t	0.356	0.437	0.342 ± 0.002
$TF-IDF_t$	0.368	0.452	0.355 ± 0.002

Table 8
Evaluation results for different methods for computing word importance scores to WAM when $M = 3$ for the BOOK dataset.

Weighting	Precision	Recall	F-measure
$TF-IDF_w$	0.368	0.452	0.355 ± 0.002
TextRank	0.345	0.424	0.332 ± 0.002
Product	0.368	0.451	0.354 ± 0.002

the resource, whereas TF_t tends to select the most popular tags, because the latter does not consider global information (the IDF_t part). Table 7 verifies the analysis, where $TF-IDF_t$ is slightly better than TF_t .

5.3.1.4. Methods for computing word importance scores. In Table 8, we show the performance of WAM applied to the BOOK dataset with different methods for computing word importance scores. From the table, we can see that there is no significant difference between $TF-IDF_w$ and the product of $TF-IDF_w$ and TextRank, and TextRank performs the worst. This performance indicates that TextRank is less competitive when measuring word importance scores, as it does not take global information into consideration.

5.3.1.5. Training data size. We also investigated the influence of training data size for WAM. As shown in Fig. 5, we increased the training data size from 8000 to 56,000 step by 8000, and performed evaluation on 4000 resources. The figure shows that: (1) when the training data size is small (e.g., 8000), WAM can still achieve good performance, and (2) when the training data size increases, the performance will be improved, but the improvement speed will be slowed down as the training data size increases. This indicates that WAM does not require a huge size dataset to achieve good performance.

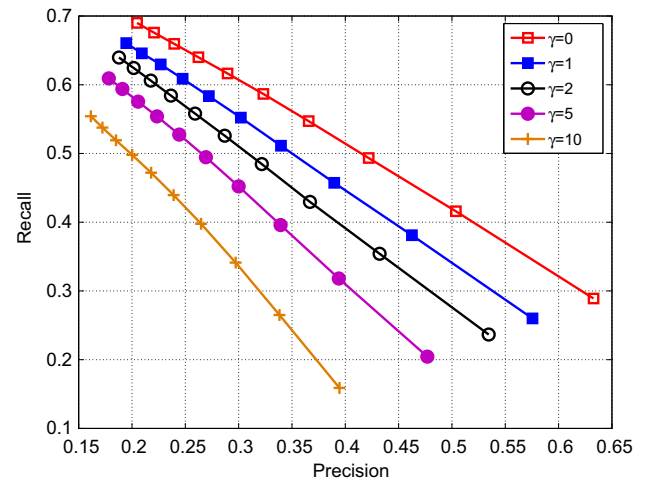


Fig. 6. Precision–recall curves of MI when the tag co-occurrence threshold γ increases from 0 to 10 on the BOOK dataset.

5.3.2. Parameter influences for MI

The parameters of the MI-based method include tag co-occurrence threshold and types of word importance scores. When investigating one parameter, we set the other parameters to be the values inducing the best performance as mentioned in Section 5.2. Similar to WAM, we only present the experimental results for the BOOK dataset for analysis.

5.3.2.1. Tag co-occurrence threshold. Fig. 6 shows the influence of the tag co-occurrence threshold for MI on the BOOK dataset. We set the tag co-occurrence threshold γ to different values. The figure shows that the MI-based method achieves the best performance with the BOOK dataset when $\gamma = 0$. This indicates that the set of tags that have low co-occurrences with a word not only contains noisy tags, but also contains proper tags that need to be suggested by the word.

5.3.2.2. Methods for computing word importance scores. In Table 9, we show the performance of MI applied to the BOOK dataset with different methods for computing word importance scores. From the table, we can see that for MI, $TF-IDF_w$ performs the best and TextRank performs the worst. It is similar to WAM, and these

Table 9

Evaluation results for different methods for computing word importance scores with MI when $M = 3$ for the BOOK dataset.

Weighting	Precision	Recall	F-measure
TF-IDF _w	0.422	0.493	0.397 ± 0.002
TextRank	0.393	0.461	0.370 ± 0.002
Product	0.407	0.475	0.382 ± 0.002

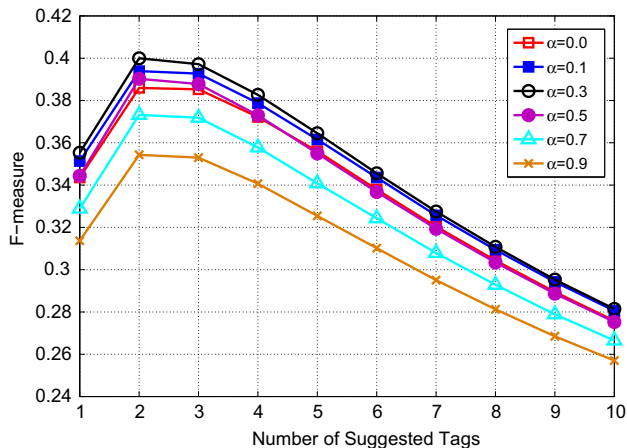


Fig. 7. F-measure of MI versus the number of suggested tags for the BOOK dataset when the self-translation parameter α ranges from 0.0 to 0.9.

results indicate that TextRank is less competitive for measuring word importance scores, as it does not take global information into consideration.

By analyzing the influences of parameters on WAM and MI, we find that the word translation model is robust to parameter variations.

5.4. Performance of emphasizing self-translation probability

In Fig. 7 we investigate the influence of self-translation parameter via the curves of F-measure of MI versus the number of suggested tags on the two datasets when the self-translation parameter α ranges from 0.0 to 0.9. As shown in Section 3.3, the parameter α controls the self-translation probabilities.

From Fig. 7, we observe that MI achieve the best performance when $\alpha = 0.2$ for the BOOK dataset, whereas $\alpha = 0.4$ on BIBTEX dataset. These results indicate that for both dataset, for one word, we need to emphasize the translation probability to itself. This is reasonable because without self-translation, a important tag may not be suggested as it does not suggest itself using the translation probabilities. We also see that for different dataset, the self-translation parameter is not a constant and varies from the need of emphasizing the words in the current document.

Finally, we tested the performance of emphasizing self-translation probability for WAM with different word translation methods for the BOOK dataset. As shown in Table 10, emphasizing the self-translation probability improves the performance of WAM (in Table 8) as applied to the BOOK dataset when using TF-IDF_w and the product as the methods for computing the word trigger powers, but decays when using TextRank. This result verifies that TF-IDF_w is the best method to measure word importance scores for WAM, which indicates that emphasizing the tags appearing in the descriptions may enhance the classification performance of the word translation method.

However, the performance when emphasizing the self-translation probability on the BIBTEX dataset decays much compared with WAM. The F-measure of emphasizing the self-translation

Table 10

The evaluation results for emphasizing the self-Translation probability on MI with different methods for computing word importance scores when $M = 3$ with the BOOK dataset.

Weighting	Precision	Recall	F-measure
TF-IDF _w	0.385	0.472	0.371 ± 0.001
TextRank	0.344	0.423	0.332 ± 0.002
Product	0.374	0.457	0.360 ± 0.001

probability is only $F = 0.229$ compared with WAM $F = 0.267$. The main reason for the decay is that the average length of descriptions in the BIBTEX dataset is too short to provide sufficient information to precisely emphasize tags, and usually emphasize wrong tags and drop correct tags.

The experimental results for emphasizing the self-translation probability suggest that, we have to analyze the characteristics of the tag suggestion systems to decide whether to emphasize the tags that appear in the corresponding descriptions. It is also worth investigating the problem when combining with collaboration-based methods for social tag suggestion.

6. Conclusions

In this paper, we present a new perspective on social tag suggestion and propose two methods to estimate translation probabilities between words in descriptions and tags. One method is the word alignment model in statistical machine translation and the other method is mutual information. Based on the translation probabilities between words and tags, we propose the word translation method. The experiments revealed that our method is effective and efficient for social tag suggestion compared with other baseline methods.

There are several open issues for further investigation:

1. Our model focuses on suggesting social tags according to the resource descriptions. We will take advantages of more social information such as user information to improve the performance of social tag suggestion.
2. Other metadata of resources (author, title, images and videos) can be also taken into account to improve the performance of social tag suggestion.
3. Our model is a supervised model which requires a large collection of annotated resources. We will explore to use large-scale unlabeled text corpora to estimate the translation probabilities between words, which could be used to enhance the estimation of translation probabilities between words in descriptions and tags in annotations.
4. In this paper we suggest each tag in isolation, without considering the correlations between tags. We will investigate the hierarchical structure and the semantic relatedness between tags to regularize the granularity of tags.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61170196 and 61202140. The authors would like to thank Peng Li for his insightful suggestions.

References

- Blei, D., & Jordan, M. (2003). Modeling annotated data. In *Proceedings of SIGIR* (pp. 127–134).
- Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *JMLR*, 3, 993–1022.

- Brown, P., Pietra, V., Pietra, S., & Mercer, R. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 263–311.
- Bundschuh, M., Yu, S., Tresp, V., Rettinger, A., Dejeri, M., & Kriegel, H. (2009). Hierarchical bayesian models for collaborative tagging systems. In *Proceedings of ICDM* (pp. 728–733).
- Fujimura, S., Fujimura, K., & Okuda, H. (2008). Blogosonomy: Autotagging any text using bloggers' knowledge. In *Proceedings of WI* (pp. 205–212).
- Herlocker, J., Konstan, J., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of SIGIR* (pp. 230–237).
- Herlocker, J., Konstan, J., Terveen, L., & Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5–53.
- Heymann, P., Ramage, D., & Garcia-Molina, H. (2008). Social tag prediction. In *Proceedings of SIGIR* (pp. 531–538).
- Iwata, T., Yamada, T., & Ueda, N. (2009). Modeling social annotation data with content relevance using a topic model. In *Proceedings of NIPS* (pp. 835–843).
- Jaschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., & Stumme, G. (2008). Tag recommendations in social bookmarking systems. *AI Communications*, 21(4), 231–247.
- Karimzadehgan, M., & Zhai, C. (2010). Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of SIGIR* (pp. 323–330).
- Katakis, I., Tsoumakas, G., & Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. ECML PKDD discovery challenge 2008, p. 75.
- Krestel, R., Fankhauser, P., & Nejdl, W. (2009). Latent dirichlet allocation for tag recommendation. In *Proceedings of ACM RecSys* (pp. 61–68).
- Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on ubiquitous information management and communication* (pp. 208–211). ACM.
- Lee, S., & Chun, A. (2007). Automatic tag recommendation for the web 2.0 blogosphere using collaborative tagging and hybrid ann semantic structures. In *Proceedings of WSEAS* (pages 88–93).
- Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of ICML* (Vol. 98, pp. 296–304).
- Liu, Z., Chen, X., & Sun, M. (2011). A simple word trigger method for social tag suggestion. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1577–1588). Association for Computational Linguistics.
- Liu, Z., Huang, W., Zheng, Y., & Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 366–376). Association for Computational Linguistics.
- Manning, C., Raghavan, P., & Schtze, H. (2008). *Introduction to information retrieval*. New York, NY, USA: Cambridge University Press.
- Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of EMNLP* (pp. 404–411).
- Mishne, G. (2006). Autotag: A collaborative approach to automated tag assignment for weblog posts. In *Proceedings of WWW* (pp. 953–954).
- Och, F., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), 19–51.
- Ohkura, T., Kiyota, Y., & Nakagawa, H. (2006). Browsing system for weblog articles based on automated folksonomy. In *Proceedings of WWW*.
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.
- Rendle, S., Balby Marinho, L., Nanopoulos, A., & Schmidt-Thieme, L. (2009). Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of KDD* (pp. 727–736).
- Resnick, P., & Varian, H. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56–58.
- Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing and management*, 24(5), 513–523.
- Si, X., Liu, Z., & Sun, M. (2010). Modeling social annotations via latent reason identification. *IEEE Intelligent Systems*.
- Si, X., & Sun, M. (2009). Tag-LDA for scalable real-time tag recommendation. *Journal of Computational Information Systems*, 6(1), 23–31.
- Wang, M., Ni, B., Hua, X.-S., & Chua, T.-S. (2012). Assistive tagging: A survey of multimedia tagging with human-computer joint exploration. *ACM Computing Surveys (CSUR)*, 44(4), 25.
- Xu, Z., Fu, Y., Mao, J., & Su, D., 2006. Towards the semantic web: Collaborative tag suggestions. In *Collaborative web tagging workshop at WWW2006*.