A Simple Word Trigger Method for Social Tag Suggestion

Zhiyuan Liu, Xinxiong Chen and **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
{lzy.thu, cxx.thu}@gmail.com, sms@tsinghua.edu.cn

Abstract

It is popular for users in Web 2.0 era to freely annotate online resources with tags. To ease the annotation process, it has been great interest in automatic tag suggestion. We propose a method to suggest tags according to the text description of a resource. By considering both the description and tags of a given resource as summaries to the resource written in two languages, we adopt word alignment models in statistical machine translation to bridge their *vocabulary gap*. Based on the translation probabilities between the words in descriptions and the tags estimated on a large set of description-tags pairs, we build a word trigger method (WTM) to suggest tags according to the words in a resource description. Experiments on real world datasets show that WTM is effective and robust compared with other methods. Moreover, WTM is relatively simple and efficient, which is practical for Web applications.

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. Table 1 shows a book entry annotated with multiple tags by users[1]. On the top of Table 1 we list the title and a short introduction of the novel "The Count of Monte Cristo". The bottom half of Table 1 shows the annotated tags, each of which is followed by a number in bracket, the total number of users who use the tag to annotate this book. Since the tags of a resource are annotated collaboratively by multiple users, we also name these tags as *social* tags. For a resource, we refer to the additional information, such as the title and introduction of a book, as *description*, and the user-annotated social tags as *annotation*.

Description
`Title:` The Count of Monte Cristo `Intro:` *The Count of Monte Cristo* is one of the most popular fictions 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), ...

Table 1: An example of social tagging. The number in the bracket after each tag is the total count of users that annotate the tag on this book.

Social tags concisely indicate the main content of the given resource, and potentially reflect user interests. Social tagging has thus been widely studied and successfully applied in recommender systems (Eck et al., 2007; Yanbe et al., 2007; Zhou et al., 2010), trend detection and tracking (Hotho et al., 2006), personalization (Wetzker et al., 2010), advertising (Mirizzi et al., 2010), etc.

The task of automatic social tag suggestion is to automatically recommend tags for a user when he/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

[1]The original record is obtained from the book review website Douban (`www.douban.com`) in Chinese. Here we translate it to English for comprehension.

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1577–1588,
Edinburgh, Scotland, UK, July 27–31, 2011.

tagging behaviors, which is playing a more and more important role in social computing and information retrieval (Wang et al., 2007).

Most online resources contain descriptions, which usually contain much resource information. For example, on a book review website, each book entry contains a title, the author(s) and an introduction of the book. Some researchers thus propose to automatically suggest tags based on resource descriptions, which are collectively known as the *content-based approach*.

One may think to suggest tags by selecting important words from descriptions. This is far from enough because descriptions and annotations are using diverse vocabularies, usually referred to as a *vocabulary gap* problem. Take the book entry in Table 1 for instance, the word "popular" used in the description contrasts the tags "classic" and "famous book" in the annotation; the word "novel" is used in the description, while most users annotate with the tag "fiction". The vocabulary gap usually reflects in two main issues:

- Some tags in the annotation do appear in the corresponding description, but they may not be statistically significant.

- Some tags may even not appear in the description.

It is not trivial to reduce the vocabulary gap and find the semantic correspondence between descriptions and annotations. By regarding both the description and the annotation as *parallel* summaries of a resource, we use word alignment models in statistical machine translation (SMT) (Brown et al., 1993) to estimate the translation probabilities between the words in descriptions and annotations. SMT has been successfully applied in many applications to bridge vocabulary gap. For detailed descriptions of related work, readers can refer to Section 2.2. In this paper, besides employing word alignment models to social tagging, we also propose a method to efficiently build description-annotation pairs for sufficient learning translation probabilities by word alignment models.

Based on the learned translation probabilities between words in descriptions and annotations, we regard the tagging behavior as a word trigger process:

1. A user reads the resource description to realize its substance by seeing some important words in the description.

2. Triggered by these important words, the user translates them into the corresponding tags, and annotates the resource with these tags.

Based on this perspective, we build a simple word trigger method (WTM) for social tag suggestion. In Fig. 1, we use a simple example to show the basic idea of using word trigger for social tag suggestion. In this figure, some words in the first sentence of the book description in Table 1 are triggered to the tags in annotation.

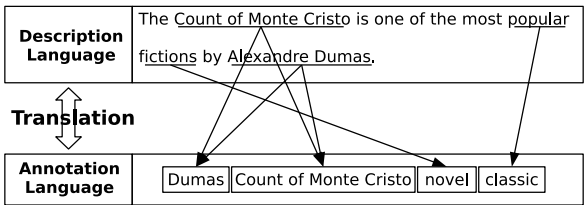


Figure 1: An example of the word trigger method for suggesting tags given a description.

2 Related Work

2.1 Social Tag Suggestion

Previous work has been proposed to automatic social tag suggestion.

Many researchers built tag suggestion systems based on *collaborative filtering* (CF) (Herlocker et al., 1999; Herlocker et al., 2004), a widely used technique in recommender systems (Resnick and Varian, 1997). These collaboration-based methods typically base their suggestions on the tagging history of the given resource and user, without considering resource descriptions. FolkRank (Jaschke et al., 2008) and Matrix Factorization (Rendle et al., 2009) are representative CF methods for social tag suggestion. Most of these methods suffer from the *cold-start problem*, i.e., they are not able to perform effective suggestions for resources that no one has annotated yet.

The content-based approach for social tag suggestion remedies 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.

Some researchers regarded social tag suggestion as a classification problem by considering each tag as a category label (Ohkura et al., 2006; Mishne, 2006; Lee and Chun, 2007; Katakis et al., 2008; Fujimura et al., 2008; Heymann et al., 2008). Various classifiers such as Naive Bayes, kNN, SVM and neural networks have been explored to solve the social tag suggestion problem.

There are two issues emerging from the classification-based methods:

- The annotations provided by users are noisy, and the classification-based methods can not handle the issue well.
- The training cost and classification cost of many classification-based methods are usually in proportion to the number of classification labels. These methods may thus 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 popularity of latent topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 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 et al., 2009; Si and Sun, 2009). Bundschus 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 aimed at finding the content-related tags for suggestion. Empirical experiments showed that CRM outperformed both classification methods and Corr-LDA (Blei and 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 topic models, such as Hierarchical Dirichlet Process (HDP) (Teh et al., 2006) and nested Chinese Restaurant Process (Blei et al., 2010), to automatically adjust the number of topics during training. Both solutions are usually computationally complicated. What is more important, topic-based methods suggest tags by measuring the topical relevance of tags and resource descriptions. The latent topics are of concept-level which are usually too general to precisely suggest those specific 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. However, TAM assumes each tag can only have at most one word as its reason. This is against the fact that a tag may be annotated triggered by multiple words in the description.

It should also be noted that social tag suggestion is different from automatic keyphrase extraction (Turney, 2000; Frank et al., 1999; Liu et al., 2009a; Liu et al., 2010b; Liu et al., 2011). Keyphrase extraction aims at selecting terms from the given document to represent the main topics of the document. On the contrary, in social tag suggestion, the suggested tags do not necessarily appear in the given resource description. We can thus regard social tag suggestion as a task of selecting appropriate tags from a controlled tag vocabulary for the given resource description.

2.2 Applications of SMT

SMT techniques have been successfully used in many tasks of information retrieval and natural language processing to bridge the vocabulary gap between two types of objects. Some typical tasks are document information retrieval (Berger and Lafferty, 1999; Murdock and Croft, 2004; Karimzadehgan and Zhai, 2010), question answering (Berger et al., 2000; Echihabi and Marcu, 2003; Soricut and Brill, 2006; Riezler et al., 2007; Surdeanu et al., 2008; Xue et al., 2008), query expansions (Riezler et al., 2007; Riezler et al., 2008; Riezler and Liu, 2010), paraphrasing (Quirk et al., 2004; Zhao et al., 2010a; Zhao et al., 2010b), summarization (Banko et al., 2000), collocation extraction (Liu et al., 2009b;

Liu et al., 2010c), keyphrase extraction (Liu et al., 2011), sentiment analysis (Dalvi et al., 2009), computational advertising (Ravi et al., 2010), and image/video annotation and retrieval (Duygulu et al., 2002; Jeon et al., 2003).

3 Word Trigger Method for Social Tag Suggestion

3.1 Method Framework

We describe the word trigger method (WTM) for social tag suggestion as a 3-stage process:

1. Preparing description-annotation pairs. Given a collection of annotated resources, we first prepare description-annotation pairs for learning translation probabilities using word alignment models.

2. Learning a translation model. 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.

3. Suggesting tags given a resource description. After building translation probabilities between words and tags, given a resource description, we first compute the trigger power of each word in the description and then suggest tags according to their translation probabilities from the triggered words.

Before introducing the method in details, we introduce the notations. In a social tagging system, a resource is denoted as $r \in R$, where R is the set of all resources. Each resource 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 $\mathbf{w}_r = \{(w_i, e_i)\}_{i=1}^{N_r}$, where e_i is the count of word w_i and N_r is the number of unique words in r. The annotation a_r of resource r is represented as $\mathbf{t}_r = \{(t_i, e_i)\}_{i=1}^{M_r}$, where e_i is the count of tag t_i and M_r is the number of unique tags for r.

3.2 Preparing Description-Annotation Pairs

Learning translation probabilities requires a parallel training dataset consisting of a number of aligned sentence pairs. We assume the description and the annotation of a resource as being written in two distinct languages. We thus prepare our parallel training dataset by pairing descriptions with annotations.

The annotation of a resource is a bag of tags with no position information. We thus select IBM Model-1 (Brown et al., 1993) for training, which does not take word position information into account on both sides for each aligned pair.

In a social tagging system, the length of a resource description is usually limited to hundreds of words. Meanwhile, it is common that some popular resources are 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. We have to deal with the length-unbalance between a resource description and its corresponding annotation for two reasons.

- 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 sentence pairs in the parallel training data set (Och and Ney, 2003).

- Moreover, the annotated tags may have different importance for the resource. It would be unfair to treat these tags without distinction.

Here 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 with comparable length with the description.

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 sampling types, including tag frequency (TF_t) within the annotation and tag-frequency inverse-resource-frequency (TF-IRF_t). Given resource r, TF_t and TF-IRF_t of tag t are defined as $\text{TF}_t = e_t / \sum_t e_t$ and $\text{TF-IRF}_t = e_t / \sum_t e_t \times \log\left(|R| / |\sum_{r \in R} I_{e_t > 0}|\right)$, where $|\sum_{r \in R} I_{e_t > 0}|$ indicates the number of resources that have been annotated with tag t.

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

3.3 Learning Translation Probabilities Using Word Alignment Models

Suppose the source language is resource description and the target language is resource annotation. 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:

$$\Pr(w_1^J|t_1^I) = \sum_{a_1^J} \Pr(w_1^J, a_1^J|t_1^I). \quad (1)$$

The alignment a_1^J also contains empty-word alignments $a_j = 0$ which align source words to the an empty word. IBM Model-1 can be trained using 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. We will learn translation models on two directions: one is regarding descriptions as the source language and annotations as the target language, and the other is in 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:

$$\Pr(t|w) \propto \Big(\lambda/\Pr{}_{d2a}(t|w)+(1-\lambda)/\Pr{}_{a2d}(t|w)\Big)^{-1}, \quad (2)$$

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

3.4 Tag Suggestion Using Triggered Words and Translation Probabilities

When given the description of a resource, we can rank tags by computing the scores:

$$\Pr(t|d = \mathbf{w}_d) = \sum_{w \in \mathbf{w}_d} \Pr(t|w) \Pr(w|d), \quad (3)$$

in which $\Pr(w|d)$ is the trigger power of the word w in the description, which indicates the importance of the word. According to the ranking scores, we can suggest the top-ranked tags to users.

Here we explore three methods to compute the trigger power of a word in a resource description: TF-IRF$_w$, TextRank and their product. TF-IRF$_w$ and TextRank are two most widely adopted methods for keyword extraction.

Similar to TF-IRF$_t$ mentioned in Section 3.2, TF-IRF$_w$ considers both the local importance (TF$_w$) and global specification (IRF$_w$).

TextRank (Mihalcea and Tarau, 2004) is a graph-based method to compute term importance. Given a resource description, TextRank first builds a term graph by connecting the terms in the description according to their semantic relations, and then run PageRank algorithm (Page et al., 1998) to measure the importance of each term in the graph. Readers can refer to (Mihalcea and Tarau, 2004) for detailed information.

We also use the product of TF-IRF$_w$ and TextRank to weight terms, which potentially takes both global information and term relations into account.

Emphasize Tags Appearing In Description for WTM (EWTM) In some social tagging systems, the tags that appear in the resource description are more likely to be selected by users for annotation. Therefore, we propose to emphasize the tags in the description by ranking tags as follows

$$\Pr(t|d) = \sum_{w \in \mathbf{w}_d} \big(\gamma I_t(w) + (1-\gamma) \Pr(t|w)\big) \Pr(w|d), \quad (4)$$

where $I_t(w)$ is an indicator function which gets value 1 when $t = w$ and 0 when $t \neq w$; and γ is the smooth factor with range $\gamma \in [0.0, 1.0]$. When $\gamma = 1.0$, it suggests tags simply according to their trigger powers within the description, while when $\gamma = 0.0$, it does not emphasize the tags appearing in the description and just suggests according to their translation probabilities. In Section 4.4, we will show the performance of EWTM.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets In our experiments, we select two real world datasets which are of diverse properties to evaluate our methods. In Table 2 we show the detailed statistical information of the two datasets.

Data	R	W	T	$\bar{N}_w$	$\bar{N}_t$
BOOK	70,000	174,748	46,150	211.6	3.5
BIBTEX	158,924	91,277	50,847	5.8	2.7

Table 2: Statistical information of two datasets. R, W, T, $\bar{N}_w$ and $\bar{N}_t$ 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.

The first dataset, denoted as BOOK, is 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, is obtained from an English online bibliography website `www.bibsonomy.org`[2]. The dataset contains the descriptions for academic papers (including the title and note for each paper) 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 tag is annotated to a resource.

Evaluation Metrics We use precision, recall and F-measure 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

$$p = \frac{|T_s \cap T_a|}{|T_s|}, \quad r = \frac{|T_s \cap T_a|}{|T_a|}, \quad F = \frac{2pr}{(p+r)}. \tag{5}$$

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

4.2 Comparing Results

Baseline Methods We select four content-based algorithms as the baselines for comparison: Naive Bayes (NB) (Manning et al., 2008), k nearest neighbor algorithm (kNN) (Manning et al., 2008), Content Relevance (CRM) model (Iwata et al., 2009) and Tag Allocation Model (TAM) (Si et al., 2010).

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

$$\Pr(t|d) \propto \Pr(t) \prod_{w \in d} \Pr(w|t). \tag{6}$$

$\Pr(t)$ is estimated by the frequency of the resources 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 recommends 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 experiments, we evaluated the performance of CRM with different K values, and here we only show the best one obtained by setting $K = 1,024$. TAM is also a generative model which 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). For comparison, we denote our method as WTM.

Complexity Analysis We compare the complexity of these methods. We denote the number of training iterations in CRM, TAM and WTM as I [3], and the number of topics in CRM as K. For the training phase, the complexity of NB is $O(R\bar{N}_w\bar{N}_t)$, kNN is $O(1)$, TAM is $O(IR\bar{N}_w\bar{N}_t)$, CRM is $O(IKR\bar{N}_w\bar{N}_t)$, and WTM is $O(IR\bar{N}_w\bar{N}_t)$[4]. When suggesting for a given resource description with length N_w, the complexity of NB is $O(N_wT)$, kNN is $O(R\bar{N}_w\bar{N}_t)$, CRM is $O(IKN_wT)$, TAM

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

[3]In fact, the numbers of iterations of the three methods are different from each other. For simplicity, here we denote them using the same notation.

[4]In more detail, the training phase of WTM contains preparing parallel training dataset with $O(R\bar{N}_t)$ and learning translation probabilities using word alignment models with $O(IR\bar{N}_w\bar{N}_t)$, where I is the number of iterations for learning translation probabilities, and $\bar{N}_t$ is the average number of tags for each resource after sampling.

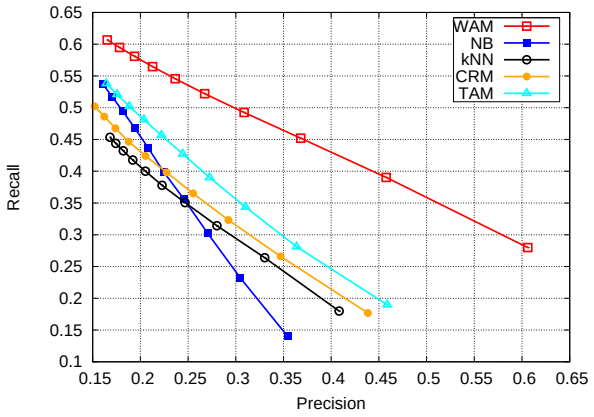
is $O(IN_wT)$ and WTM is $O(N_wT)$. From the analysis, we can see that WTM is a relatively simple method for both training and suggestion. This is especially valuable because WTM also shows good effectiveness for tag suggestion compared with other methods as we will shown later.

Parameter Settings We use GIZA++ (Och and Ney, 2003)[5] as IBM Model-1 to learn translation probabilities using description-annotation pairs for WTM. The experimental results of WTM are obtained by setting parameters as follows: tag weighting type as TF-IRF$_t$, length ratio $\delta = 1$, harmonic factor $\lambda = 0.5$ and the type of word trigger strength as TF-IRF$_w$. The influence of parameters to WTM can be found in Section 4.3.

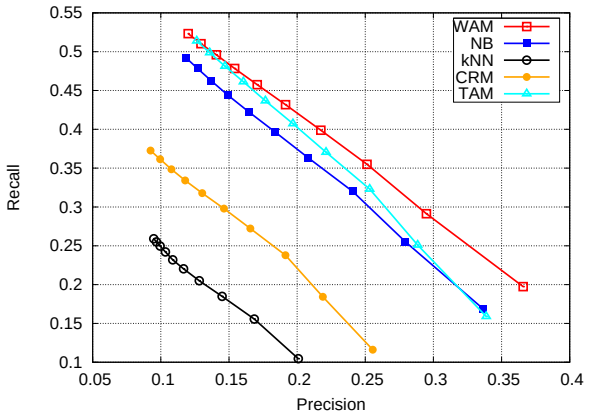
Experiment Results and Analysis In Fig. 2 we show the precision-recall curves of NB, kNN, CRM and WTM on two datasets. Each point of a precision-recall curve represents different numbers of suggested tags from $M = 1$ (bottom right, with higher precision and lower recall) to $M = 10$ (upper left, with higher recall but lower precision) respectively. The closer the curve to the upper right, the better the overall performance of the method. From Fig. 2, we observe that:

- WTM consistently performs the best on both datasets. This indicates that WTM is robust and effective for tag suggestion.
- The advantage of WTM is more significant on the BOOK dataset. The reason is that WTM can take a good advantage of annotation count information of tags compared to other methods.
- The average length of resource descriptions is short in the BIBTEX dataset, which makes it difficult to determine the trigger powers of words. But even on the BIBTEX dataset with no count information of tags, WTM still outperforms other methods especially when recommending first several tags.

To further demonstrate the performance of WTM and other baseline methods, in Table 3 we show the



(a) BOOK



(b) BIBTEX

Figure 2: Performance comparison between NB, kNN, CRM, TAM and WTM on two datasets.

precision, recall and F-measure of NB, kNN, CRM, TAM and WTM on BOOK dataset when suggesting $M = 3$ tags[6]. Due to the limit of space, we only show the variance of F-measure. In fact, WTM achieves its best performance when $M = 2$, where the F-measure of WTM is 0.370, outperforming both CRM ($F = 0.263$) and TAM ($F = 0.277$) by about 10%.

An Example In Table 4 we show top 10 tags suggested by NB, CRM, TAM and WTM for the book in Table 1. The number in bracket after the name of each method is the count of correctly suggested tags. The correctly suggested tags are marked in bold face. We select not to show

[5]GIZA++ is freely available on `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.

[6]We select to show this number because it is near the average number of tags for BOOK dataset

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
WTM	**0.368**	**0.452**	$\mathbf{0.355 \pm 0.002}$

Table 3: Comparing results of NB, kNN, CRM, TAM and WTM on BOOK dataset when suggesting $M = 3$ tags.

the results of kNN 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 general tags such as "novel", "literature", "classic" and "France", and fail in suggesting specific tags such as "Alexandre Dumas" and "Count of Monte Cristo". On the contrary, WTM succeeds in suggesting both general and specific tags related to the book.

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
WTM (+7): **novel**, **Alexandre Dumas**, **history**, **Count of Monte Cristo**, **foreign literature**, biography, **suspense**, comic, America, **France**

Table 4: Top 10 tags suggested by NB, CRM, TAM and WTM for the book in Table 1.

In Table 5, we list four important words (using TF-IRF$_w$ as weighting metric) of the description and their corresponding tags with the highest translation probabilities. The values in brackets are the probability of tag t given word w, $\Pr(t|w)$. For each word, we eliminated the tags with the probability less than 0.1. We can see that the translation probabilities can map the words in descriptions to their semantically corresponding tags in annotations.

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

4.3 Parameter Influences

We explore the parameter influences to WTM for social tag suggestion. The parameters include harmonic factor, length ratio, tag weighting types, and types of word trigger strength. When investigating one parameter, we set other parameters to be the values inducing the best performance as mentioned in Section 4.2. Finally, we also investigate the influence of training data size for suggestion performance. In experiments we find that WTM reveals similar trends on both the BOOK dataset and the BIBTEX dataset. We thus only show the experimental results on the BOOK dataset for analysis.

Harmonic Factor In Fig. 3 we investigate the influence of harmonic factor via the curves of F-measure of WTM 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 3.3, 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 harmonic mean, the performance is consistently better, especially when λ ranges from 0.2 to 0.6. This is reasonable because IBM Model-1 constrains that only the term in source language can be aligned to multiple terms in target language, which makes the translation probability learned by a single model be asymmetric.

Length Ratio Fig. 4 shows the influence of length ratios 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.,

$\delta = 10$)[7].

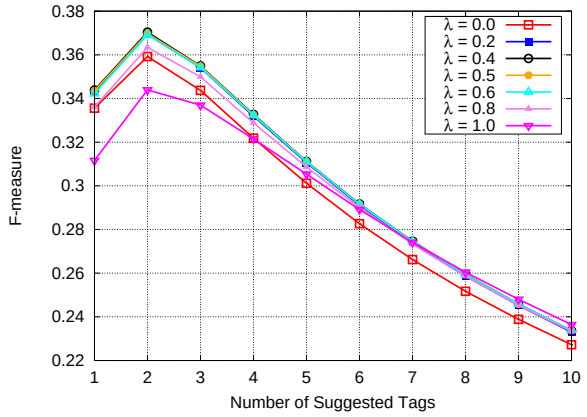


Figure 3: F-measure of WTM versus the number of suggested tags on the BOOK dataset when harmonic factor λ ranges from 0.0 to 1.0.

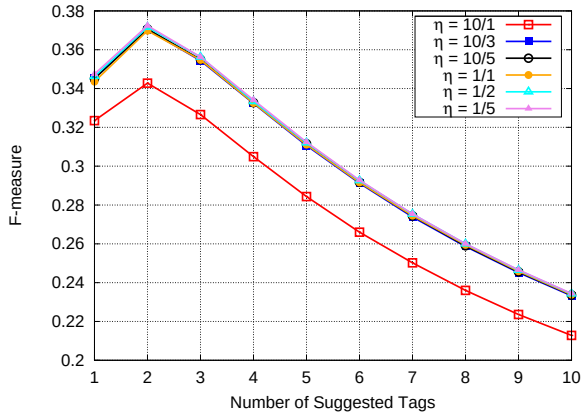


Figure 4: F-measure of WTM versus the number of suggested tags on the BOOK dataset when length ratio δ ranges from $10/1$ to $1/5$.

Tag Weighting Types The influence of two weighting types, TF_t and TF-IRF_t, on social tag suggestion when $M = 3$ on the BOOK dataset is shown in Table 6. TF-IRF_t tends to select the tags more specific to the resource while TF_t tends to select the most popular tags, because the latter does not consider global information (the IRF_t part).

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

Table 6 verifies the analysis, where TF-IRF_t is slightly better than TF_t.

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

Table 6: Evaluation results for different tag weighting types when $M = 3$ on the BOOK dataset.

Methods for Computing Word Trigger Power In Table 7, we show the performance of social tag suggestions on the BOOK dataset with different methods for computing word trigger power. From the table, we can see that there is not significant difference between TF-IRF_w and the product of TF-IRF_w and TextRank, while TextRank itself performs the worst. This indicates that TextRank is less competitive to measure word trigger power since it does not take global information into consideration.

Weighting	Precision	Recall	F-measure
TF-IRF_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

Table 7: Evaluation results for different methods for computing word trigger powers when $M = 3$ on the BOOK dataset.

Training Data Size We investigate the influence of training data size for social tag suggestion. As shown in Fig. 5, we increased the training data size from $8,000$ to $56,000$ step by $8,000$, and carried out evaluation on $4,000$ resources. The figure shows that:

- When the training data size is small (e.g., $8,000$), WTM can still achieve good suggestion performance.
- As the training data size increases, the performance of WTM improves, while the improvement speed declines.

The observation indicates that WTM does not require huge-size dataset to achieve good performance.

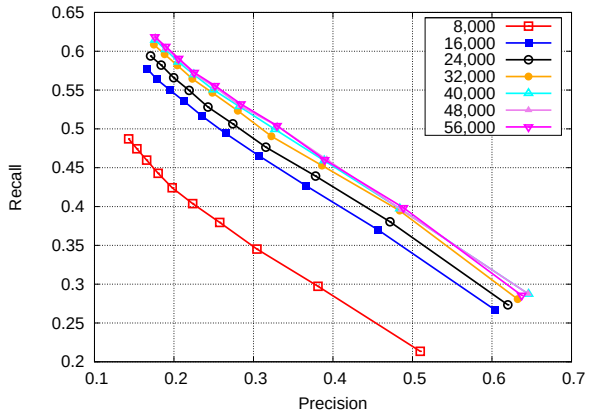


Figure 5: Precision-recall curves when the training data size increases from $8,000$ thousand to $56,000$ thousand on the BOOK dataset.

Conclusion By analyzing the influences of parameters on WTM, we find that WTM is robust to parameter variations.

4.4 Performance of EWTM

At the end of this section, we investigate the performance of EWTM for social tag suggestion. Here we simply set the smooth factor $\gamma = 0.5$.

As shown in Table 8, EWTM improves the performance of WTM (in Table 7) on the BOOK dataset when using TF-IRF$_w$ and the product as the methods for computing the word trigger powers, but decays when using TextRank. This verifies that TF-IRF$_w$ is the best method to measure word trigger powers for WTM. Table 8 indicates that emphasizing the tags appearing in the descriptions may enhance the suggestion power of the word trigger method.

Weighting	Precision	Recall	F-measure
TF-IRF$_w$	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

Table 8: The evaluation results of EWTM with different methods for computing word trigger powers when $M = 3$ on the BOOK dataset.

However, the performance of EWTM on the BIBTEX dataset decays much compared to WTM. The F-measure of EWTM is only $F = 0.229$ compared with WTM $F = 0.267$. The main reason of the decay is that: the resource descriptions in the BIBTEX dataset are usually too short to provide sufficient information to precisely emphasize tags. In this case, EWTM may emphasize wrong tags and drop correct tags.

The experimental results on EWTM suggest that, the performance of EWTM is heavily influenced by the length of resource descriptions. Therefore, we have to analyze the characteristics of social tagging systems to decide whether to emphasize the tags that appear in the corresponding resource descriptions.

As future work, we will investigate the influence of the smooth factor γ to EWTM. It is also worth to investigate the problem when combining with collaboration-based methods for social tag suggestion.

5 Conclusion and Future Work

In this paper, we present a new perspective to social tagging and propose the word trigger method for social tag suggestion based on word alignment in statistical machine translation. Experiments show that our method is effective and efficient for social tag suggestion compared to other baselines.

There are still several open problems that should be further investigated:

1. We can exploit other word alignment methods like log-linear models (Liu et al., 2010a) for social tag suggestion.

2. We will ensemble WTM with other content-based and collaboration-based methods to build a practical social tag suggestion system.

3. WTM and EWTM can only suggest the tags that have appeared in translation models. In future, we plan to incorporate keyphrase extraction in social tag suggestion to make it suggest more appropriate tags not only from translation models but also from the resource descriptions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 60873174. The authors would like to thank Peng Li for his insightful suggestions and thank the anonymous reviewers for their helpful comments.

References

M. Banko, V.O. Mittal, and M.J. Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of ACL*, pages 318–325.

A. Berger and J. Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of SIGIR*, pages 222–229.

A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of SIGIR*, pages 192–199.

D.M. Blei and M.I. Jordan. 2003. Modeling annotated data. In *Proceedings of SIGIR*, pages 127–134.

D.M. Blei, A.Y. Ng, and M.I. Jordan. 2003. Latent dirichlet allocation. *JMLR*, 3:993–1022.

D.M. Blei, T.L. Griffiths, and M.I. Jordan. 2010. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7.

P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

M. Bundschus, S. Yu, V. Tresp, A. Rettinger, M. Dejori, and H.P. Kriegel. 2009. Hierarchical bayesian models for collaborative tagging systems. In *Proceedings of ICDM*, pages 728–733.

N. Dalvi, R. Kumar, B. Pang, and A. Tomkins. 2009. A translation model for matching reviews to objects. In *Proceeding of CIKM*, pages 167–176.

P. Duygulu, K. Barnard, J. De Freitas, and D. Forsyth. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *Proceedings of ECCV*, pages 97–112.

A. Echihabi and D. Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of ACL*, pages 16–23.

D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. 2007. Automatic generation of social tags for music recommendation. In *Proceedings of NIPS*, pages 385–392.

E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceedings of IJCAI*, pages 668–673.

S. Fujimura, KO Fujimura, and H. Okuda. 2008. Blogosonomy: Autotagging any text using bloggers' knowledge. In *Proceedings of WI*, pages 205–212.

J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of SIGIR*, pages 230–237.

J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53.

P. Heymann, D. Ramage, and H. Garcia-Molina. 2008. Social tag prediction. In *Proceedings of SIGIR*, pages 531–538.

A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. 2006. Trend detection in folksonomies. *Semantic Multimedia*, pages 56–70.

T. Iwata, T. Yamada, and N. Ueda. 2009. Modeling social annotation data with content relevance using a topic model. In *Proceedings of NIPS*, pages 835–843.

R. Jaschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. 2008. Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247.

J. Jeon, V. Lavrenko, and R. Manmatha. 2003. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of SIGIR*, pages 119–126.

M. Karimzadehgan and C.X. Zhai. 2010. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of SIGIR*, pages 323–330.

I. Katakis, G. Tsoumakas, and I. Vlahavas. 2008. Multilabel text classification for automated tag suggestion. *ECML PKDD Discovery Challenge 2008*, page 75.

R. Krestel, P. Fankhauser, and W. Nejdl. 2009. Latent dirichlet allocation for tag recommendation. In *Proceedings of ACM RecSys*, pages 61–68.

S.O.K. Lee and A.H.W. Chun. 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.

Z. Liu, P. Li, Y. Zheng, and M. Sun. 2009a. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of EMNLP*, pages 257–266.

Z. Liu, H. Wang, H. Wu, and S. Li. 2009b. Collocation extraction using monolingual word alignment method. In *Proceedings of EMNLP*, pages 487–495.

Y. Liu, Q. Liu, and S. Lin. 2010a. Discriminative word alignment by linear modeling. *Computational Linguistics*, 36(3):303–339.

Z. Liu, W. Huang, Y. Zheng, and M. Sun. 2010b. Automatic keyphrase extraction via topic decomposition. In *Proceedings of EMNLP*, pages 366–376.

Z. Liu, H. Wang, H. Wu, and S. Li. 2010c. Improving statistical machine translation with monolingual collocation. In *Proceedings of ACL*, pages 825–833.

Z. Liu, X. Chen, Y. Zheng, and M. Sun. 2011. Automatic keyphrase extraction by bridging vocabulary gap. In *Proceedings of CoNLL*, pages 135–144.

C.D. Manning, P. Raghavan, and H. Schtze. 2008. *Introduction to information retrieval*. Cambridge University Press New York, NY, USA.

R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP*, pages 404–411.

R. Mirizzi, A. Ragone, T. Di Noia, and E. Di Sciascio. 2010. Semantic tags generation and retrieval for online advertising. In *Proceedings of CIKM*, pages 1089–1098.

G. Mishne. 2006. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proceedings of WWW*, pages 953–954.

V. Murdock and W.B. Croft. 2004. Simple translation models for sentence retrieval in factoid question answering. In *Proceedings of SIGIR*.

F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

T. Ohkura, Y. Kiyota, and H. Nakagawa. 2006. Browsing system for weblog articles based on automated folksonomy. In *Proceedings of WWW*.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford Digital Library Technologies Project*.

C. Quirk, C. Brockett, and W. Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, volume 149.

S. Ravi, A. Broder, E. Gabrilovich, V. Josifovski, S. Pandey, and B. Pang. 2010. Automatic generation of bid phrases for online advertising. In *Proceedings of WSDM*, pages 341–350.

S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. 2009. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of KDD*, pages 727–736.

P. Resnick and H.R. Varian. 1997. Recommender systems. *Communications of the ACM*, 40(3):56–58.

S. Riezler and Y. Liu. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics*, 36(3):569–582.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proccedings of ACL*, pages 464–471.

S. Riezler, Y. Liu, and A. Vasserman. 2008. Translating queries into snippets for improved query expansion. In *Proceedings of COLING*, pages 737–744.

X. Si and M. Sun. 2009. Tag-LDA for scalable real-time tag recommendation. *Journal of Computational Information Systems*, 6(1):23–31.

X. Si, Z. Liu, and M. Sun. 2010. Modeling social annotations via latent reason identification. *IEEE Intelligent Systems*, 25(6):42 – 49.

R. Soricut and E. Brill. 2006. Automatic question answering using the web: Beyond the factoid. *Information Retrieval*, 9(2):191–206.

M. Surdeanu, M. Ciaramita, and H. Zaragoza. 2008. Learning to rank answers on large online qa collections. In *Proceedings of ACL*, pages 719–727.

Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

P.D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

F.Y. Wang, K.M. Carley, D. Zeng, and W. Mao. 2007. Social computing: From social informatics to social intelligence. *IEEE Intelligent Systems*, 22(2):79–83.

R. Wetzker, C. Zimmermann, C. Bauckhage, and S. Albayrak. 2010. I tag, you tag: translating tags for advanced user models. In *Proceedings of WSDM*, pages 71–80.

X. Xue, J. Jeon, and W.B. Croft. 2008. Retrieval models for question and answer archives. In *Proceedings of SIGIR*, pages 475–482.

Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. 2007. Can social bookmarking enhance search in the web? In *Proceedings of JCDL*, pages 107–116.

S. Zhao, H. Wang, X. Lan, and T. Liu. 2010a. Leveraging multiple mt engines for paraphrase generation. In *Proceedings of COLING*, pages 1326–1334.

S. Zhao, H. Wang, and T. Liu. 2010b. Paraphrasing with search engine query logs. In *Proceedings of COLING*, pages 1317–1325.

T.C. Zhou, H. Ma, M.R. Lyu, and I. King. 2010. UserRec: A user recommendation framework in social tagging systems. In *Proceedings of AAAI*, pages 1486–1491.