Automatic Web Tagging and Person Tagging Using Language Models

Qiaozhu Mei[1] and Yi Zhang[2]

[1] University of Illinois at Urbana-Champaign, USA
qmei2@uiuc.edu
[2] University of California at Santa Cruz, USA
yiz@soe.ucsc.edu

Abstract. Social bookmarking has become an important web2.0 application recently, which is concerned with the dual user behavior to search - tagging. Although social bookmarking websites, e.g., Del.icio.us, has been attracting much attentions, many research problems in social tagging has not been well addressed in literature. In this paper, we formally define the problem of social bookmark suggestion, and propose a probabilistic language modeling approach to automatically label the target web documents with meaningful phrases. The probabilistic language models trained from social tagging logs are used to automatically generate tags which capture the semantics of web documents. We also adapt the modeling approach to label internet users. Empirical experiments show that our approach is effective to suggest meaningful tags for web documents as well as web users.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Text Mining

General Terms: Algorithms

Keywords: social bookmarking, tag suggestion, language modeling, automatic labeling

1 Introduction

The explosive growth of user generated data has attracted much attention from business and academic research. Many new paradigms of user behaviors are supported by various novel web2.0 applications. Social tagging systems let user manually enter labels to tag an object/page. The tags can use it later to help the user re-find the object through search, let the user expand the knowledge about the object and share the customized object with other people. Social tagging systems, such as Delicious, My Web 2.0, Flickr, YouTube, have been very successful and attracted hundreds of million users.

Unlike research problems on search engines which have been well studied, many problems on social tagging have not been well addressed in literature. However, tagging is a time consuming process for the user. Finding appropriate

words or phrases to tag an object is an expensive mental process. It could be tedious for a mobile web users to assign tags using inconvenient input methods.

Instead of waiting for a user to find and input the appropriate words to tag an object, we propose to automatically recommend tags for social bookmarking systems. The user only needs to choose from recommended tags, a process that requires much less cognitive effort than traditional tagging. In particular, we formalize the tag suggestion problem as a ranking problem and propose a new probabilistic language model to rank meaningful tags, including words or phrases, for bookmarks. Besides, we adapt the probabilistic language model to tag users. The user tags can viewed as recommended queries for the user to search documents. They can also be used as meta data about the users, which could be beneficial for people search or person recommendation. The effectiveness of the proposed techniques are demonstrated on data collected from del.icio.us.

The application of the tag suggestion technique is not limited to social bookmarking systems. There are many other scenarios that such a technique can be found useful. For example, in an online advertising systems like Google AdWords [3], it is very important for a business provider to select appropriate tags for their website, which are expected to be used as search queries by the search users. One could imagine that the automatic tag suggestion method could be useful for such a task, or for people tagging in online social networks.

The rest of this paper is organized as follows. In Section 2, we formally define the problem of bookmark suggestion as a ranking problem. We then propose an probabilistic approach to tag suggestion with automatic labeling of web documents, in Section 3. We present our experiments, related work, and our conclusions in Section 4, 5, and 6 respectively.

2 Problem Definition

To the best of our knowledge, the problem of social bookmark suggestion is not well defined in existing literature. In this section, we give a formal definition of the research problem. We begin with the definition of a few key concepts.

Definition 1. ***Web document****. A web document, denoted as d, is a web page associated with a unique Uniform Resource Locator (URL). Please note that the content of a web document could change over time.*

Definition 2. ***Tag****. A tag, denoted as t, is a short text segment selected by a user to label a web document d. A natural instantiation of a tag is a single word, or a phrase. We further assume that there is a vocabulary V of all possible tags.*

Definition 3. ***Bookmark****. A bookmark, r, is a sequence of tags $r[T] = t_1 t_2 ... t_l$, selected by a user $r[U]$ to mark a web document $r[D]$ in a social bookmarking system. We further define the log of a social bookmarking system as a set of bookmarks.*

[3] https://adwords.google.com/select/Signup1/index.html

Based on the definitions above, one may easily see the duality of retrieval and tagging. Indeed, the essential goal of social bookmarking is to allow a user to quickly retrieve a tagged web document based on the tags he used in the bookmark. In other words, a tag is likely to be used in the future as a query.

The problem of retrieval (or web search if the targets are web documents) is usually cast as a ranking problem - given a query q, rank the documents based on a scoring function $f(d, q)$. Following this line, we define the problem of tag suggestion also as a ranking problem.

Definition 4. ***Tag Suggestion****. Given a web document d, the problem of tag suggestion is to generate a ranked list of tags by some scoring function $f(t, d)$.*

Apparently, this problem is challenging - in some aspects, even more challenging than retrieval. First, unlike in retrieval where the collection of documents is fixed, the target set of potential tags is unknown. How to guarantee that the tags are meaningful to a user is challenging. Second, there is usually a gap between the vocabulary used by real users and the vocabulary of a web document. For example, Chinese users may use Chinese words to tag an English web page. Moreover, in a social bookmarking system, the tags that a user selected may be highly correlated to what other users used to tag the web document.

3 Tag Suggestion by Automatic Labeling Tagging Logs

In this section, we introduce a probabilistic approach to automatically generate tag suggestions by labeling language models estimated from tagging logs.

3.1 Candidate Tag Generation

Given a web document, the first step of tag suggestion is to find a set of candidate tags. The most straight forward way is to use the words and phrases in the content of the web document. However, such a method suffers from several problems. Unlike a commercial search engine, a social bookmarking system usually do not keep an index of the actual content of web pages. The vocabulary used in a web document could also be quite different from the tags used by real users. A better method of candidate tag generation should be not relying on the actual content of a web page.

Tag Extraction from Tagging Logs: One treatment is to generate such candidate tags from the tagging logs of a social bookmarking system. The tags used in such tagging logs are selected by real users instead of web documents themselves, thus are more likely to be adopted by the coming users. Figure 1 presents an example bookmark in a social bookmarking system.

From the example above, we can see that a bookmark has the following characteristics: 1) in tagging logs, each bookmark is usually a sequence of tags instead of a single tag; 2) people usually use meaningful phrases rather than

URL	"http://englishcaster.com/bobrob/"		
User	so****	Time	2007-02-10
Bookmark	Blog Bob ESL English Funny Ideas Learning Lessons Podcast Rob		
URL	"http://www.speakoz.com/english-directory/lesson-plans/"		
User	so****	Time	2007-02-10
Bookmark	Australia ESL English Learning Lesson OZ Plans Speak		

Fig. 1. Example Bookmarks in Tagging Logs

single words as tags (e.g., "funny ideas," "learning lessons," "ESL English"); 3) there is usually no explicit segmentation of tags in a bookmark; and 4) the sequence of tags doesn't follow syntax rules.

Based on such characteristics of bookmarks, there is thus a need to extract actual tags from bookmarks. One natural solution is to simply use every word in the bookmarks as candidate tags. However, in reality people usually use meaningful phrases rather than single words as tags. Based on this assumption, we introduce a method to extract meaningful phrases from bookmarks. Since bookmarks usually don't follow syntax rules, we could not use NLP parsers to extract natural phrases. Instead, we extract phrases by ranking word ngrams based on statistical tests. The basic idea is that if the words in an ngram tend to co-occur with each other, the ngram is more likely to be an n-word phrase.

Many methods have been proposed to test whether an ngram is a meaningful collocation/phrase [3, 16, 1, 9]. Some relies on statistical measures such as mutual information [3] and others rely on hypothesis testing. The null hypothesis is that "the words in an ngram are independent", and there are different test statistics to test the significance of violating the null hypothesis. A well used hypothesis testing method showing good performance on phrase extraction is the Student's T-Test [9]. Based on the occurrence statistics, the Student's T-Test will assign a score called T-score to each ngram. We then extract ngrams with top-k T-score as candidate tags.

Tag Extraction from Outside Resource: Another method is to rely on an outside dictionary to extract candidate tags. Again, we need to guarantee that the candidate tags are likely to be used as tags by real web users. Therefore, it is reasonable to extract tags from a collection of user generated text collection rather than from other sources such as a dictionary. In our experiments, we use the titles of every entry in Wikipedia[4] as candidate tags.

3.2 A Probabilistic Approach to Tag Ranking

Once the candidate tags are extracted, the next step is to rank the tags based on their relevance to a web document. Based on the duality of search and tagging, a straight forward solution is to borrow the ranking methods in retrieval. In

[4] http://en.wikipedia.org

retrieval problems, queries and documents are represented with the same model, and a similarity based scoring function is introduced based on such a model. Language modeling has been widely adopted in information retrieval recently [], which leads to good retrieval performance as well as many extensions. In this work, we follow the language modeling retrieval framework, and represent a tag and a web document both with a unigram language model.

Formally, we extract a unigram language model, or a multinomial distribution of words from a web document d and a candidate tag t, denoted as $\{p(w|d)\}$ and $\{p(w|t)\}$ respectively. Since we do not rely on the actual content of d, we alternatively estimate $p(w|d)$ based on the social tagging logs. Specifically, we have

$$p(w|d) = \frac{\sum_r c(w, r[T]) \cdot \mathbb{I}[r[D] = d]}{\sum_{w'} \sum_r c(w', r[T]) \cdot \mathbb{I}[r[D] = d]} \tag{1}$$

where $\mathbb{I}[S]$ is an indicator function which is set to 1 if the statement S is true and 0 otherwise, and $c(w, r[T])$ is the occurrence of word w in the tags $r[T]$.

Once we estimated a language model for d, the problem is reduced to selecting tags to label such a multinomial model of words. The easiest way is apparently using words with largest $p(w|d)$. This method may have problem because it is usually hard to interpret the meaning of a distribution of words from just the top words because the top words are usually obscure and only partially captures the information encoded by the whole distribution [12]. Instead, we expect labels that could cover the semantics conveyed by the whole distribution (e.g., a distribution with top words like "tree", "search", "DFS", "prune", "construct" is better to be labeled as "tree algorithms" rather than "tree"). We then follow [12] and present a probabilistic approach to automatically label the document language model $p(w|d)$. The basic idea is that if we can also extract a language model from a tag t, we could use the Kullback-Leibler (KL) divergence to score each candidate tag. Specifically,

$$f(t, d) = -D(d||t) = \sum_w p(w|d) \log \frac{p(w|t)}{p(w|d)}. \tag{2}$$

How to estimate $p(w|t)$ is however tricker. The simplest way is to estimate $p(w|t)$ based on the word frequency in a tag. However, a tag is usually too short (e.g., 1 or 2 words) to estimate a reliable language model. If we embed such an estimate in Equation 2, we need to smooth the tag language model so that $\forall w, p(w|t) > 0$ [17]. When a tag is very short, such a smoothed language model would not be trustable. Indeed, in this case $f(t, d)$ will be similar to count the occurrence of t in all bookmarks of d.

We need to find a reliable language model $p(w|t)$ in an alternatively way. One possibility is to approximate $p(w|t)$ from the collection of tagging logs C, and estimate a distribution $p(w|t, C)$ to substitute $p(w|t)$. Similar to [12], we can rewrite Equation 2 as

$$f(t, d) = \sum_w p(w|d) \log \frac{p(w|t, C)}{p(w|C)} - \sum_w p(w|d) \log \frac{p(w|d)}{p(w|C)} - \sum_w p(w|d) \log \frac{p(w|t, C)}{p(w|t)}$$

$$= \sum_w p(w|d) \log \frac{p(w,t|C)}{p(w|C)p(t|C)} - D(d||C) + Bias(t,C). \quad (3)$$

From this rewriting, we see that the scoring function can be decomposed into three components. The second component $D(d||C)$ is irrelevant to t, and can be ignored in ranking. The third component $-\sum_w p(w|d) \log \frac{p(w|t,C)}{p(w|t)}$ can be interpreted as the bias of using C to approximate the unknown language model $p(w|t)$. When the t and C are from the same domain (e.g., if we use C to extract candidate tags), we can fairly assume that such bias is ignorable. Therefore, we have

$$f(t,d) \stackrel{\text{rank}}{=} \sum_w p(w|d) \log \frac{p(w,t|C)}{p(w|C)p(t|C)} = E_d[PMI(w,t|C)]. \quad (4)$$

Please note that $\log \frac{p(w,t|C)}{p(w|C)p(t|C)}$ is actually the pointwise mutual information of t and w conditional on the tagging logs. $f(t,d)$ can thus be interpreted with the expected mutual information of t and a word in the document language model. Estimating $PMI(w,t|C)$ is straight forward, since we use efficiently find $p(w,t|C), P(w|C)$ and $P(t|C)$ that maximize the likelihood that each word is cooccurring with t in the tagging records. Specifically, we have

$$p(w,t|C) = \frac{\sum_{r \in C} c(w, r[T]) \cdot \mathbb{I}[t \in r[T]]}{\sum_{w'} \sum_{t'} \sum_{r \in C} c(w', r[T]) \cdot \mathbb{I}[t' \in r[T]]} \quad (5)$$

$$p(w|C) = \frac{\sum_{r \in C} c(w, r[T])}{\sum_{w'} \sum_{r} c(w', r[T])} \quad (6)$$

$$p(t|C) = \frac{\sum_{r \in C} \mathbb{I}[t \in r[T]]}{\sum_{t'} \sum_{r \in C} \mathbb{I}[t' \in r[T]]}. \quad (7)$$

Once the candidate tags are extracted, all the mutual information $PMI(w,t|C)$ can be computed and stored offline. This also improves the efficiency in ranking. Computing the expectation of the mutual information could still be time consuming if the vocabulary is large. To further improve the runtime efficiency, we ignore all the mutual information where $PMI(w,t|C) < 0$. We then select the tags with largest $f(t,d)$ as the suggested tags to a document d.

3.3 Further discussion: tagging users and beyond

So far, we have presented a probabilistic approach to tag suggestion for social bookmarking systems. We cast this problem as automatic labeling of web document language model estimated from the tagging logs.

Since eventually we are labeling language models, this method can be applied to suggest tags for other objects besides web documents, as long as a corresponding language model can be estimated for such an object from tagging logs. For example, can we tag users instead of web pages using similar techniques? The answer is yes. Specifically, one can first estimate a user language model by

$$p(w|u) = \frac{\sum_r c(w, r[T]) \cdot \mathbb{I}[r[U] = u]}{\sum_{w'} \sum_r c(w', r[T]) \cdot \mathbb{I}[r[U] = u]}. \quad (8)$$

The same ranking function can be used to generate labels for this user language model, by replacing $p(w|d)$ with $p(w|u)$ in Equation 4.

Similarly, by taking time into consideration, one can also suggest tags for a web document in different time periods, by replacing $p(w|d)$ with $p(w|d, time)$. Another interesting variation is personalized tag suggestion. One can either use a user specific candidate tag set, or compute the pointwise mutual information in a different manner, e.g., replace $PMI(w, t|C)$ with $PMI(w, t|C, u)$. We leave such variations for future work.

4 Experiments

In this section, we use empirical experiments to show the effectiveness of our proposed methods.

4.1 Data

We collect two-week tagging records from Del.icio.us[5], a well known social bookmarking website. From each tagging record, we extract a bookmark, one URL, and a user ID. The basic statistics of this dataset is given in Table 1.

Table 1. Basic Statistics of Del.icio.us Tagging Logs

Dataset	Time Span	Bookmarks	Distinct Tagging Words	Distinct Users
Del.icio.us	02/13/2007 - 02/26/2007	579,652	111,381	20,138

To test different ways of candidate tag generation, we also collect all the titles of entries in Wikipedia, from the data snapshot of 10/18/2007. There are in total 5,836,166 entries extracted.

4.2 Candidate Tags

We explore three different ways of candidate tag generation. For the first method, we simply use single words in the tagging logs as candidate tags. For the second method, we extract significant bigrams from tagging logs using Student's T-Test. This was done using the N-gram Statistics Package [1]. We select the top 15,000 bigrams with the largest T-Score as the candidate tags. The top ranked bigrams are presented in Table 2. For the third method, we use all titles of Wikipedia entries as candidate tags.

It is easy to see that most of the top bigrams extracted from the tagging logs are meaningful. However, they could overfit the log data, where some words are user specific (e.g., webdesign), and some bigrams contain redundant words (e.g., photo photography). Such problem does not show in Wikipedia entries. However, only 48k such entry titles appear in the tagging logs, out of 5,836k.

[5] http://del.icio.us/

Table 2. Top Bigrams from Tagging Logs

Bigrams with Highest T-score			
css design	software tools	web webdesign	mac osx
programming reference	web web2.0	art design	rails ruby
mp3 music	tools web	photo photography	photography photos

4.3 Tagging Web Documents

The first experiment designed is to suggest tags for web documents. We select web documents with the largest number of bookmarks in the tagging log collection, and automatically suggest tags for them. The results are presented in Table 3.

Table 3. Tag Suggestions for Web Documents

URLs	LM $p(w\|d)$	Tag = Word	Tag = Bigram	Tag = Wiki Entry
http://pipes.yahoo.com/ 386 bookmarks	yahoo	**pipes**	**feeds_mashup**	**pipes**
	rss	feeds	**mashup_pipes**	yahoo
	web2.0	yahoo	web2.0_yahoo	mashup
	mashup	mashup	rss_web2.0	rss
	feeds	rss	mashup_rss	syndication
	programming	syndication	api_feeds	mashups
	pipes	mashups	pipes_programming	blog_feeds
http://www.miniajax.com/ 349 bookmarks	ajax	ajax	**ajax_code**	ajax
	javascript	dhtml	code_javascript	dhtml
	web2.0	javascript	javascript_ajax	javascript
	webdesign	moo.fx	javascript_web2.0	moo.fx
	programming	dragdrop	css_ajax	**javascript_library**
	code	phototype	programming_web2.0	javascript_framework
	webdev	autosuggest	javascript_programming	ajax_framework
http://kuler.adobe.com/ 158 bookmarks	color	**color**	**adobe_color**	color
	design	**colour**	**color_design**	colour
	webdesign	**palette**	color_colour	palette
	tools	colorscheme	color_colors	**web_color**
	adobe	colours	colour_desgin	colours
	graphics	picker	inspiration_palete	cor
	flash	cor	webdesign_color	rgb
http://www.youtube.com/ watch?v=6gmP4nk0EOE 157 bookmarks	**web2.0**	**youtube**	**xml_youtube**	**internet_video**
	video	**revver**	**web2.0_youtube**	youtube
	youtube	vodcast	video_web2.0	**revver**
	web	primer	web2.0_xml	research_video
	internet	comunidad	online_presentation	vodcast
	xml	participation	social_video	primer
	community	ethnograpy	youtube_video	p2p_TV
http://www.picnik.com/ 149 bookmarks	photo	photo	**editor_flickr**	photo
	photography	resize	**editor_online**	resize
	tools	flickr	**online_photo**	flickr
	editor	editor	editor_photo	editor
	online	edit	photography_tools	edit
	web2.0	editing	photo_tools	editing
	flickr	crop	editor_image	crop

We present the top words in the document language model $p(w|d)$ estimated from the tagging logs in the second column. The right three columns present system generated tag suggestions, using single words, significant bigrams, and wikipedia titles as candidate tags, respectively.

There are several interesting discoveries from Table 3. First, as expected, if we simply use top words in $p(w|d)$ as tag suggestions, it is hard to interpret the semantics of the web document. Such a simple method favors frequent terms, such as "web2.0", which appears in the top words of the language models for many web documents. It is thus not desirable as a bookmark, because when a user uses it in the future trying to retrieve the documents, he has to spend much extra effort to target the web document from the many documents bookmarked with "web2.0". Other examples are like "web", "webdesign", "tools", "code", "online", and "internet". It is also hard to interpret the semantics of the web document just from the top words.

When we use the labeling based method, we get much better tag suggestions. In column 3, we can see that "pipes" is apparently a better tag than "yahoo" because it captures the meaning of the url "http://pipes.yahoo.com" more precisely. "youtube" is also more precise than all other words in column 2 for the url "http://www.youtube.com/watch?v=6gmP4nk0EOE", which is a video on youtube. "Palette" is a very interesting generalization of the meaning of "http://kuler.adobe.com/," which does not appear in the top words in $p(w|d)$. This is because the method we introduce tries to capture the meaning of the whole language model (i.e., the expectation of similarity of a tag to all the words), which thus generates more precise tags.

However, the tags generated are sometimes still obscure or not meaningful. For example, "color," "photo," and "editor" are obscure as tags, and "ajax," "pipes," "feeds" are ambiguous which could mean quite different concepts. Some words are also too domain specific and not so meaningful to the common audience (e.g., "dhtml," "comunidad"). All this is because single words are used as candidate tags. When phrases (significant bigrams, wikipedia entry names) are used as candidate tags, we see that the system generates much more understandable suggestions.

When we use statistical significant bigrams as candidate tags, the suggestions are much more precise and interpretable. "Ajax code," "mashup pipes," and "api feeds" remove the ambiguity of single words. "Photography tools," "editor flickr," and "color design" are also more precise than "tools," "photo," "editor," and "design."

There is also a concern for this method because the statistical ngram may overfit the text collection, and the extracted phrases are not "real" phrases. Indeed, we see examples like "xml youtube," "adobe color," "color colors," and "css ajax," which are good tags but not real phrases. In real life, people may not use such expressions. By using wikipedia entry names as candidate tags, the suggestions are guaranteed to be meaningful concepts and understandable to general audience (e.g., "blog feeds, " "javascript library," "internet video," etc).

4.4 Tagging Users

A different scenario is to suggest tags for a web user. Similarly, we select 10 users with the largest number of bookmarks in the tagging log collection, and automatically suggest tags for them. The results are presented in Table 4.

Table 4. Tag Suggestions for Web Users

Users	LM $p(w\|d)$	Tag = Bigram	Tag = Wiki Entry
User1	photography	**art_photography**	**art_photography**
	art	photography_portraits	photoblog
	portraits	**digital_flickr**	portraits
	tools	photoblog_photography	photography
	web	art_photo	landscapes
	design	flickr_photography	flickr
	geek	weblog_wordpress	**art_contest**
User2	humor	geek_hack	**network_programming**
	programming	**humor_programming**	tweak
	photography	hack_hacking	hacking
	blog	networking_programming	security
	webdesign	geek_html	**geek_humor**
	security	geek_hacking	sysadmin
	funny	reference_security	**digitalcamera**
User3	games	**arg_games**	arg
	arg	games_puzzles	**games_research**
	tools	**games_internet**	games
	programming	arg_code	puzzles
	sudoku	games_sudoku	storytelling
	cryptography	**code_generator**	**code_generator**
	software	**community_games**	**community_games**
User4	web	rubyonrails_web	javascript
	reference	**css_development**	css
	css	**brower_development**	webdev
	development	development_editor	xhtml
	rubyonrails	**development_forum**	dhtml
	tools	**development_firefox**	css3
	design	**javascript_tools**	dom

There are also interesting findings from the tag suggestions for web users. Ideally, a tag that best matches a user's preference will be suggested to him, through which he could access web documents that other people bookmarked with this tag. The preference of a user is presented with a language model estimated from his own tags in column 2. We see that our algorithm suggests interesting tags to the user, presented in column 3 and 4. Tag "art photography" perfect matches user 1's interests. If there's tags like "digital flickr" and "art contest", he is also likely to be interested. This also indicates an opportunity of personalized online advertisements.

The interests of user 2 are actually a mixture of several themes. From column 4, we clearly see that "network programming" and "geek humor" are good suggestions to such themes. However, if there is a tag "humor programming" from other users (although looks weird in reality), which perfectly matches different aspects of his interests, he is very likely to explore such a tag.

Similarly, we see that user 3 likes games and programming related content, and user 4 likes web development. Our methods suggest very highly relevant and understandable tags to them.

5 Related Work

Social bookmarking systems, which attracts a large number of users and also generates large volumes of tagging logs, has been introducing opportunities and

challenges to the research of web/text mining. Recently, researchers have started to realize the importance of social bookmarking. This leads to the exploration of tagging logs in different ways [5, 10, 4, 8, 7, 15]. Most work are focusing on utilizing social tags, instead of suggesting tags. Folksonomy [10], tagging visualization [4], and spam detection for tagging system [8] are some of such examples. [2] utilizes social tags to help summarization. [7] explores search and ranking in tagging systems. [15] first uses tagging logs to help web search, and [6] gives an empirical justification of helping search with tagging logs. [11] introduced a duality hypothesis of search and tagging, which gives a theoretical justification of using tags to help search tasks. However, none of this work explores the problem of suggesting tags for web documents, or for web users.

To the best of our knowledge, automatic bookmark suggestion is not well addressed in existing literature. The only work we are aware of is collaborative tag suggestion described in [14]. They discussed the desirable properties of suggested tags, however their tagging approach is not based on any probabilistic models and rather ad hoc.

The probabilistic language modeling framework for tagging is motivated by the well known language modeling approach in the information retrieval community. In particular, [12] has proposed to assign meaningful labels to multinomial topic models. We adapted the technique to the novel problem of tag suggestion, and generate meaningful tags for web documents and web users.

There is also early work on suggesting index terms for library documents [13]. However, all such work are based on content of documents, and is not appropriate for social bookmarking systems, where the content of web pages are hard to keep track of but rich tagging log is available.

6 Conclusions

In this work, we formally define the problem of tag suggestion for social bookmarking systems, and present a probabilistic approach to automatically generate and rank meaningful tags for web documents and web users. Empirical experiments show that our proposed methods are effective to extract relevant and meaningful tag suggestions. Such a technique could be applied to other interesting mining problems, such as ad term suggestion for online advertisement systems, and people tagging in social network applications. There are quite a few potential future directions, such as tag suggestion over time, personalized tag suggestion, and collaborative tag suggestion are all among the good examples. Another line of future work is to design a way to quantitative evaluate tag suggestion algorithms.

References

1. S. Banerjee and T. Pedersen. The design, implementation, and use of the ngram statistics package. pages 370–381, 2003.

2. O. Boydell and B. Smyth. From social bookmarking to social summarization: an experiment in community-based summary generation. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 42–51, 2007.
3. K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, 1990.
4. M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proceedings of the 15th international conference on World Wide Web*, pages 193–202, 2006.
5. S. Golder and B. A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 2006.
6. P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of the international conference on Web search and web data mining*, pages 195–206, 2008.
7. A. Hotho, R. Jasschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In Y. Sure and J. Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011, pages 411–426, 2006.
8. G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 57–64, 2007.
9. C. D. Manning and H. Schtze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA, 1999.
10. C. Marlow, M. Naaman, D. Boyd, and M. Davis. Position paper, tagging, taxonomy, flickr, article, toread. In *Proceedings of Hypertext*, pages 31–40, 2006.
11. Q. Mei, J. Jiang, H. Su, and C. Zhai. Search and tagging: Two sides of the same coin? In *UIUC Technical Report*, 2007.
12. Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *Proceedings of KDD '07*, pages 490–499, 2007.
13. B. R. Schatz, E. H. Johnson, P. A. Cochrane, and H. Chen. Interactive term suggestion for users of digital libraries: using subject thesauri and co-occurrence lists for information retrieval. In *Proceedings of the first ACM international conference on Digital libraries*, pages 126–133, 1996.
14. Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, May 2006.
15. Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web? In *JCDL*, pages 107–116, 2007.
16. C. Zhai. Fast statistical parsing of noun phrases for document indexing. In *Proceedings of the fifth conference on Applied natural language processing*, pages 312–319, 1997.
17. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, 2001.