Tag2Word: Using Tags to Generate Words for Content Based Tag Recommendation

Yong Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China
wy@smail.nju.edu.cn

Yuan Yao
State Key Laboratory for Novel Software Technology, Nanjing University, China
y.yao@nju.edu.cn

Feng Xu
State Key Laboratory for Novel Software Technology, Nanjing University, China
xf@nju.edu.cn

Hanghang Tong
Arizona State University, USA
hanghang.tong@asu.edu

Jian Lu
State Key Laboratory for Novel Software Technology, Nanjing University, China
lj@nju.edu.cn

ABSTRACT

Tag recommendation is helpful for the categorization and searching of online content. Existing tag recommendation methods can be divided into collaborative filtering methods and content based methods. In this paper, we put our focus on the content based tag recommendation due to its wider applicability. Our key observation is the *tag-content co-occurrence*, i.e., many tags have appeared multiple times in the corresponding content. Based on this observation, we propose a generative model (Tag2Word), where we generate the words based on the tag-word distribution as well as the tag itself. Experimental evaluations on real data sets demonstrate that the proposed method outperforms several existing methods in terms of recommendation accuracy, while enjoying linear scalability.

Keywords

tag recommendation; tag-content co-occurrence; generative model

1. INTRODUCTION

Tags usually indicate the keywords that help people to describe the online content, and therefore allow better information organization and retrieval. However, over 50% online content lacks tag information or even does not have tags at all [13]. Additionally, it is often painstaking for users (even the content creators) to manually tag the online content, especially under many situations where the users are not certain about what the appropriate tags are. Therefore, tag recommendation is necessary to automatically provide suitable tags for online content.

CIKM '16, October 24–28, 2016, Indianpolis, IN, USA.
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2983323.2983682

Roughly, existing tag recommendation methods can be divided into collaborative filtering methods and content based methods. Collaborative filtering methods aim to provide subjective tag recommendations based on users' historical behavior. On the other hand, content based methods take the content as input, and therefore have a wider applicability (see Section 5 for a review).

In this work, we put our focus on the content based tag recommendation. Our key observation is the *tag-content co-occurrence* phenomenon that widely exists in many online content sites. Figure 1 gives several examples from different websites. As we can see from the figures, many tags have appeared multiple times in the corresponding content. However, such co-occurrence is largely ignored by existing work.

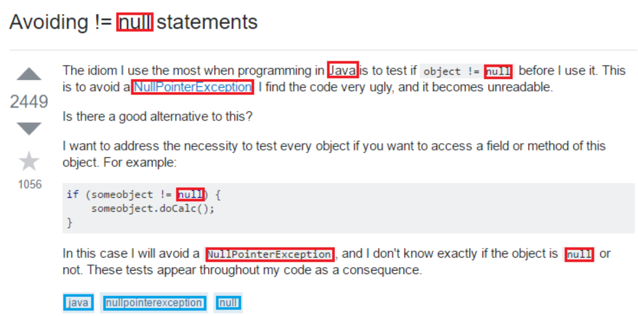
In this paper, we propose a model Tag2Word to leverage the tag-content co-occurrence phenomenon. We take a generative view by generating the content words based on the tag-word distribution[1] and the tag itself. In other words, when generating a word, we directly use the tag as the word with a probability (which can be learned for different domains), and sample the word from the tag-word distribution otherwise.

The main contributions of this paper include:

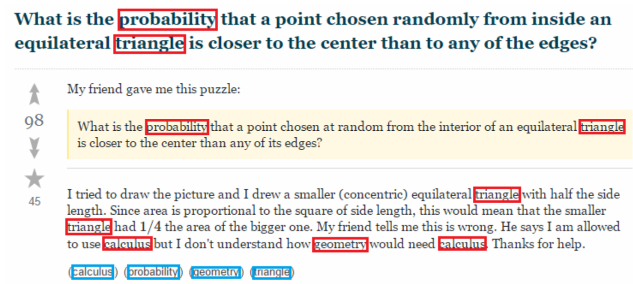
- A generative model for content based tag recommendation. The model makes use the tag-content co-occurrence observation. Typically, we use the title and the body of a post as the content (Tag2Word). To speed up, we can simply use the title as content (Tag2Word_0).
- Experimental evaluations on the two data sets demonstrating the effectiveness and efficiency of our proposed methods. For example, compared with the existing best competitors, the proposed methods can lead up to 15.0% improvement in terms of prediction accuracy, while enjoying linear scalability.

The rest of the paper is organized as follows. Section 2 presents the problem definition. Section 3 describes the proposed method. Section 4 presents the experimental eval-

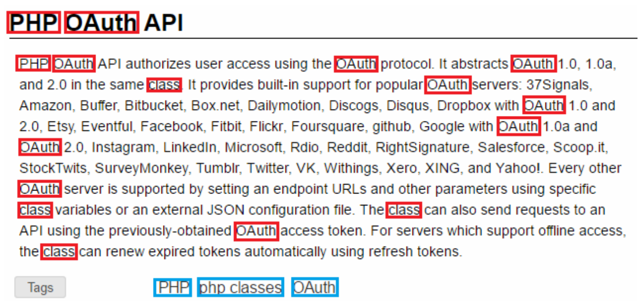
[1] The tag-word distribution or the topic-word distribution can be learned by topic models.



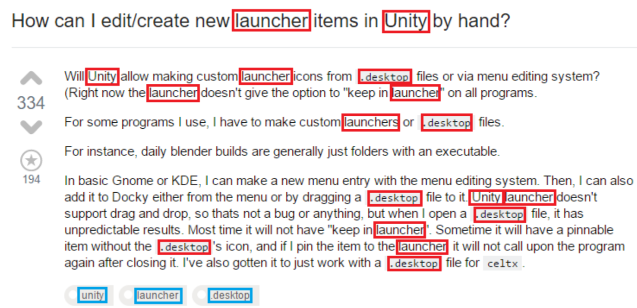
(a) Stack Overflow



(b) Mathematics Stack Exchange



(c) Freecode



(d) Ask ubuntu

Figure 1: Tag-content co-occurrence examples on different websites.

Table 1: Symbols.

Symbol	Definition and Description
D	Collection of documents
V	Vocabulary
T	Tag space
M	Total number of documents
K	Total number of latent topics
$\overrightarrow{W}_d$	List of words in document d
$\overrightarrow{\Lambda}_d$	List of tags in document d
N_d	Number of words in document d

uations. Section 5 reviews related work, and Section 6 concludes the paper.

2. PROBLEM DEFINITION

In this section, we present the problem statement. Table 1 lists the main symbols used throughout this paper. We use D to stand for a collection of input documents[2]. Each document d contains a list of words $\overrightarrow{W}_d$ and a list of tags $\overrightarrow{\Lambda}_d$. All the words form the vocabulary V, and all the tags form the tag space T. We denote K as the total number of latent topics in the input documents.

With these notations, we define the tag recommendation problem as

PROBLEM 1. *Tag Recommendation Problem*

Given: *(1) a collection of documents D, where each document $d \in D$ has its own words $\overrightarrow{W}_d$ and tags $\overrightarrow{\Lambda}_d$, and*

(2) a new document $d_{new} \notin D$ which only contains words $\overrightarrow{W}_{d_{new}}$;

Find: *the estimated list of tags for the new document d_{new}.*

In the above problem definition, although not explicitly stated, words in $\overrightarrow{\Lambda}_d$ may have appeared multiple times in $\overrightarrow{W}_d$ for a given document d. In addition, the observation holds when we only use the words in the title as the content. Identifying and exploiting such co-occurrence is the main difference between our work and the existing work on this problem.

3. THE TAG2WORD MODEL

In this section, we present the proposed Tag2Word model for Problem 1. We use generative models to incorporate the tag-content co-occurrence, as it is a more natural way compared to the discriminative models. Figure 2 shows the graphical representation for Tag2Word.

As we can see from the figure, there are three parts in Tag2Word:

- First, Tag2Word builds on the LDA model [2] to generate words for documents. For each document, LDA assumes that it has several latent topics (θ). Words are generated from a specific topic (z) and the topic-word distributions (Φ).

- Second, following the LLDA model [17], we assume that the tags Λ determine the latent topics during the generative process, and constrain that tag and latent topic are one-one correspondent. That is, each tag is

[2]In this paper, we interchangeably use 'document' and 'post' as both of them indicate the online content that we aim to recommend tags for.

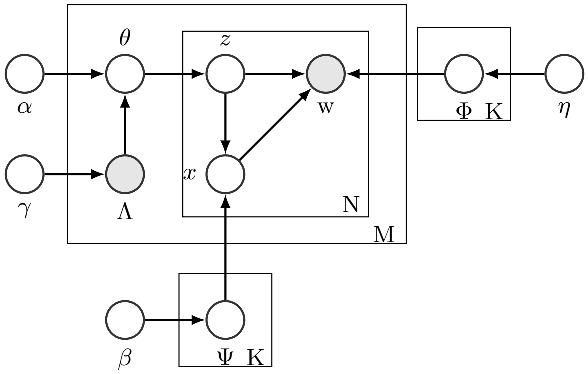


Figure 2: Graphical representation for Tag2Word.

Table 2: Generative process of Tag2Word.

1	$For\ each\ topic\ k \in [1, K]$
2	$\quad Generate\ \overrightarrow{\Phi}_k \sim Dir(\overrightarrow{\eta})$
3	$\quad Generate\ \Psi_k \sim Beta(\overrightarrow{\beta})$
4	$For\ each\ document\ d \in [1, M]$
5	$\quad For\ each\ topic\ k \in [1, K]$
6	$\quad\quad Generate\ \Lambda_{d,k} \in \{0, 1\} \sim Bernoulli(\gamma_k)$
7	$\quad Generate\ \overrightarrow{\alpha}_d = \overrightarrow{\Lambda}_d \circ \overrightarrow{\alpha}$
8	$\quad Generate\ \overrightarrow{\theta}_d \sim Dir(\overrightarrow{\alpha}_d)$
9	$\quad For\ each\ word\ i \in [1, N_d]$
10	$\quad\quad Generate\ z_i \sim Mult(\overrightarrow{\theta}_d)$
11	$\quad\quad Generate\ x_i \in \{0, 1\} \sim Bernoulli(\Psi_{z_i})$
12	$\quad\quad if\ x_i = 1$
13	$\quad\quad\quad Generate\ w_i \sim Identity(z_i)$
14	$\quad\quad else$
15	$\quad\quad\quad Generate\ w_i \sim Mult(\overrightarrow{\Phi}_{z_i})$

associated with one topic, and the topic number of a document is the same with its tag number[3].

- Third, to make use of the tag-content co-occurrence, we further add a latent variable x to indicate the probability that the word w is generated by the tag itself (z) or by the tag-word distribution (Φ). The latent variable x is sampled from a Bernoulli distribution (Ψ), and it is dependent on the specific topic (z), i.e., different topics may have different probabilities.

The generative process for our model is summarized in Table 2. For each topic k, Step 2 draws a multinomial topic-word distribution $\overrightarrow{\Phi}_k$ from a Dirichlet prior $\overrightarrow{\eta}$, and Step 3 draws a Bernoulli distribution Ψ_k from a Beta prior $\overrightarrow{\beta}$. Here, Ψ_k indicates the probability to directly use the tag as the generated word for tag/topic k. For each document d, a multinomial distribution $\overrightarrow{\theta}_d$ is drawn over restricted topics that correspond to its tags $\overrightarrow{\Lambda}_d$[4] (Steps 5-8). In Step 7, we compute the Hardamard product between $\overrightarrow{\Lambda}_d$ and $\overrightarrow{\alpha}$, so that the topic assignment z_i for each word in document d is limited within its own tags. For each word i, we use a latent variable x_i to determine where it is generated from. When x_i equals 1, the word is directly generated using the tag z_i (Steps 12-13). Otherwise, if x_i equals 0, the word is generated from the multinomial distributions $\overrightarrow{\Phi}_{z_i}$ for the topic/tag (Steps 14-15).

To solve the model, we can develop a Gibbs sampling algorithm to train the parameters. However, Gibbs sampling is inherently stochastic and unstable when iterations are not enough. On the other hand, it is noticed that the CVB0 learning algorithm [1] converges faster and more stable than the Gibbs sampling algorithm [16]. Therefore, we choose to build our learning algorithm based on CVB0 approximation algorithm. The details are omitted for brevity.

4. EXPERIMENTS

In this section, we present our experimental evaluations. The experiments are designed to answer the following questions:

- *Effectiveness*: How accurate is the proposed algorithm for tag recommendation?
- *Efficiency*: How scalable is the proposed algorithm?

4.1 Experimental Setup

We study two data sets of Stack Overflow (SO) and Mathematics Stack Exchange (Math). Both data sets are officially published and publicly available[5]. For each data set, it contains question posts and their corresponding tags. Each question post contains a title and a body. We need some preprocessing on the data sets. For the posts, we remove the stopwords and some low frequency words to reduce noise. We deliberately keep those words that are tags (e.g., C or VB). All the remaining words are then stemmed. For the tags, we remove some low frequency tags as they are seldom used. Table 3 shows the statistics of the two preprocessed data sets.

For these two data sets, we randomly select 90% posts as training data and use the rest as test data. Since some compared methods are computationally prohibitive on the whole SO data, we also randomly sample subsets (SO-10K and SO-100K) of the whole SO data to compare their effectiveness results. For the three hyper-parameters, we fix α, η, and β to $50/K$, 0.01, and 0.1, respectively.

As to evaluation metrics, we adopt Recall@n for effectiveness comparison. The reason is that finding all the useful tags in the recommendation list is important for tag recommendation tasks [19]. As to the list size n, we choose $n = 5$ and $n = 10$ as such choices will not cause many burdens to the users. Recall@n is defined as follows

$$Recall@n = \frac{1}{M} \sum_{i=1}^{M} \frac{hit(n)_i}{tag_i} \quad (1)$$

where M is the number of posts in the data, $hit(n)_i$ is the number of tags that have been successfully recommended in the top-n ranked list, and tag_i is the number of actual tags of the i^{th} post.

For efficiency, we simply report the wall-clock time of the proposed algorithm. All the experiments were run on a machine with eight 3.4GHz Intel Cores and 32GB memory.

[3]The total number of topics is the same with the total number of tags, i.e., $K = |T|$.

[4]The tags $\overrightarrow{\Lambda}_d$ in document d are observed variables in the model, and the prior γ is unused. We include it for completeness.

[5]http://blog.stackoverflow.com/tags/cc-wiki-dump/

Table 3: Statistics of the Datasets

Dataset	# of posts	Post vocabulary size	# of tags	# of average words per post
Math	19950	7705	461	54
SO	3350978	9357	1035	81

Table 4: Effective Comparisons on SO-10K data. Higher is better. Tag2Word outperforms all the compared methods.

Methods	Recall@5	Recall@10
LLDA	0.47805	0.58870
Link-LDA	0.33651	0.43010
MATAR	0.48988	0.55168
Snaff	0.38248	0.48103
Maxide	0.38995	0.46815
Tag2Word	0.56330	0.67707
Tag2Word_0	0.58538	0.65870

Table 5: Effective Comparisons on SO-100K data. Higher is better. Tag2Word outperforms all the compared methods.

Methods	Recall@5	Recall@10
LLDA	0.54254	0.64804
Link-LDA	0.32907	0.42514
MATAR	0.53774	0.59125
Snaff	0.45165	0.54921
Maxide	0.50998	0.62361
Tag2Word	0.60939	0.70339
Tag2Word_0	0.63607	0.71477

Table 6: Effective Comparisons on whole SO data. Higher is better. Tag2Word outperforms all the compared methods.

Methods	Recall@5	Recall@10
LLDA	0.55660	0.66422
Link-LDA	0.33847	0.43429
MATAR	-	-
Snaff	0.46402	0.58363
Maxide	-	-
Tag2Word	0.62169	0.71036
Tag2Word_0	0.64548	0.72682

4.2 Experiemental Results

Next, we present the experimental results.

4.2.1 *Empirical Study*

We first empirically study the tag-content co-occurrence in the two data sets. We find that more than 70% tags have appeared in the content of the SO data, while only less than 30% tags have appeared in the content of the Math data. Moreover, there are over 30% posts in the SO data that have all tags appeared in the content. We also study the degree of tag-content co-occurrence when only the title of the post is used as content. We found that 65.3% posts in SO data and 15.4% posts in Math data have at least one tag appeared in the title.

Overall, we find that the tag-content co-occurrence exists in both data sets, although the co-occurrence degree may be different in different domains. Additionally, tags may appear in both the title and the body of the content.

4.2.2 *Effectiveness Results*

For effectiveness, we compare the proposed Tag2Word with several existing methods including LLDA [17], Link-LDA [3], MATAR [9], Snaff [10], and Maxide [31]. In the compared methods, LLDA and Link-LDA are topic models, MATAR and Maxide are multi-label learning methods, and Snaff is a hybrid method. The results are shown in Table 4 - 7. In the tables, we show the results of both Tag2Word and Tag2Word_0, where Tag2Word takes both title and body as content and Tag2Word_0 takes only title as content. On the whole SO data, we do not show the results of Maxide and MATAR as they are computationally prohibitive (e.g., cannot return results in 24 hours).

There are several observations from the tables. First, Tag2Word outperforms all the compared existing methods on all the data sets. For example, on SO-10K data, Tag2Word improves its best competitors (MATAR and LLDA, respectively) by 14.9% wrt Recall@5 and by 15.0% wrt Recall@10. On Math data, Tag2Word is 3.7% and 4.1% better than the best competitor LLDA for Recall@5 and Recall@10, respectively. This result indicates that the tag-content co-occurrence indeed can help improve the recommendation accuracy, especially considering that LLDA can be a special case of Tag2Word if we delete the tag-content co-occurrence considerations in the model. Second, we find that the improvement of Tag2Word is more significant on the SO data than the Math data. This is probably due to the fact that the tag-content co-occurrence degree is higher on SO than that on Math. Third, comparing the results on three SO data, we can find that the recommendation accuracy improves as the training data size increases. Finally, Tag2Word_0 can already achieve better results than the existing methods on the SO data. Tag2Word_0 can even outperform Tag2Word in some cases. This is due to the fact that even when only the title is considered, the tag-content co-occurrence degree is already very high on the SO data. This means that we can recommend tags solely based on the title of the content, if there are plenty tags appeared in the title of the content.

4.2.3 *Efficiency Results*

Next, we study the scalability of the proposed algorithm in the training stage. We vary the size of the training data on the SO data, and report the results of wall-clock time in Figure 3. Similar results are observed on Math data, and we omit the figures for brevity.

As we can see from the figure, both Tag2Word and Tag2Word_0 scale linearly wrt the size of training data (the number of posts), which is also consistent with our algorithm analysis in Section 4.2. Additionally, Tag2Word_0 runs much faster than Tag2Word (around 9x faster), as it only involves the title as input.

We also compare the prediction time for a new post at the

Table 7: Effective Comparisons on Math data. Higher is better. Tag2Word outperforms all the compared methods.

Methods	Recall@5	Recall@10
LLDA	0.58859	0.68992
Link-LDA	0.43850	0.58259
MATAR	0.55271	0.65750
Snaff	0.48833	0.57042
Maxide	0.52436	0.64942
Tag2Word	0.61057	0.71797
Tag2Word_0	0.59352	0.68529

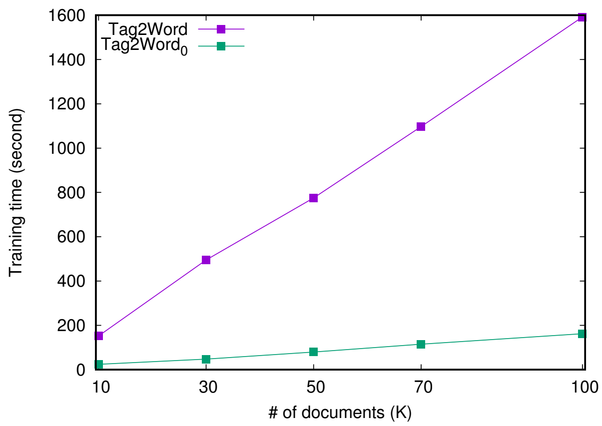


Figure 3: Scalability of the proposed algorithm2. Both Tag2Word and Tag2Word_0 scale linearly wrt the data size.

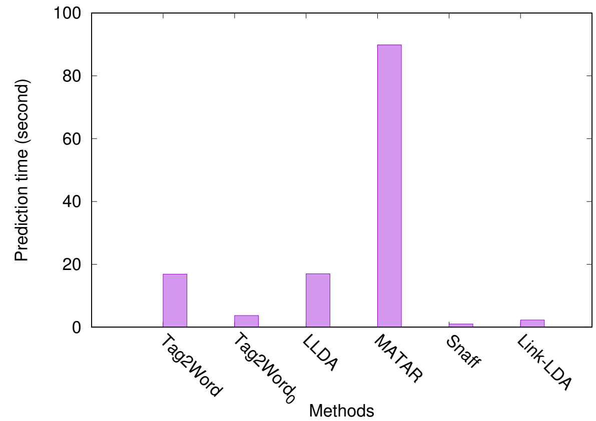


Figure 4: Response time comparison. Tag2Word_0 has a favorable response speed.

response stage of different methods. The results on Math data are showed in Figure 4. Similar results are observed on the SO data. As we can see, both Tag2Word and Tag2Word_0 can make predictions within 20 seconds. Tag2Word and LLDA have comparable response speed. The response time of MATAR is long as it adopts the lazy strategy. Snaff and Link-LDA are faster, andTag2Word_0 can have close response time with them.

5. RELATED WORK

In this section, we briefly review the related work. We roughly divide existing tag recommendation methods into collaborative filtering method and content-based method.

The key insight of collaborative filtering method is to employ the tagging histories (i.e., user-item-tag tuples). For example, Symeonidis et al. [26] model users, items, and tags into 3-order tensors and use high-order singular value decomposition to recommend tags; Rendle et al. [19, 20] further model the pairwise rankings into tensor factorization; Fang et al. [4] propose a non-linear tensor factorization method via Gaussian kernel; Feng et al. [5] model a social tagging system as a multi-type graph, and recommend tag by learning the weights of nodes and edges in the graph. Other examples in this category include [7, 23, 24, 6]. Methods in this class are more suitable to recommend a list of personalized tags for a fixed set of items, and they are not able to recommend tags for new content.

In contrast to collaborative filtering method, content based method takes the content as input, and therefore could be used to recommend tags for new content. For example, Sood et al. [25] and Mishne [14] leverage previous tags associated with similar content to recommend tags for new content. Murfi et al. [15] first use keyword extraction to filter candidate tags and then apply non-negative matrix factorization for tag recommendation; Wang et al. [28] also extract keywords, and then apply association rules to recommend tags. Erosheva et al. [3] extends LDA by mixing the generation of tags and words; Ramage et al. [17] also extends LDA by constraining the one-one correspondence between tags and latent topics. Other examples include [8, 18, 21, 29, 22]. Our method falls into this category of content based method. Different from the above work, our main observation is from the tag-content co-occurrence.

Recently, there are also some other lines of research about tag recommendation. For example, Lops et al. [12] design a hybrid tag recommender that combines the collaborative filtering and content based methods. Liu et al. [11] explore locations to recommend tags for photos. Xia et al. [30] combine several components for software information sites. Wang et al. [27] adopt a deep learning model and combine probabilistic matrix factorization to find effective and compact content representation for tag recommendation.

6. CONCLUSIONS

In this paper, we have proposed a content based tag recommendation model Tag2Word as well as its variant Tag2Word_0. Our key observation is that the tags usually appear as regular words in the content. Then, the proposed model takes a generative view to incorporate such observation to improve the recommendation accuracy. Experimental evaluations on two real data sets show that the proposed methods can lead up to 15.0% improvement over the best competitors in terms of prediction accuracy, while enjoying linear scalability in the training stage.

7. ACKNOWLEDGMENTS

This work is supported by the National Basic Research Program of China (No. 2015CB352202) and the National Natural Science Foundation of China (No. 91318301). This work is partially supported by the National Science Foundation under Grant No. IIS1017415, by DTRA under the grant number HDTRA1-16-0017, by Army Research Office under the contract number W911NF-16-1-0168, by National Institutes of Health under the grant number R01LM011986, Region II University Transportation Center under the project number 49997-33 25 and a Baidu gift.

8. REFERENCES

[1] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *UAI*, pages 27–34, 2009.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[3] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5220–5227, 2004.

[4] X. Fang, R. Pan, G. Cao, X. He, and W. Dai. Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. In *AAAO*, 2015.

[5] W. Feng and J. Wang. Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In *KDD*, pages 1276–1284, 2012.

[6] S. Hamouda and N. Wanas. Put-tag: personalized user-centric tag recommendation for social bookmarking systems. *Social network analysis and mining*, 1(4):377–385, 2011.

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

[8] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *RecSys*, pages 61–68, 2009.

[9] L. Li, Y. Yao, F. Xu, and J. Lu. Matar: Keywords enhanced multi-label learning for tag recommendation. In *APWeb*, pages 268–279, 2015.

[10] M. Lipczak and E. Milios. Learning in efficient tag recommendation. In *RecSys*, pages 167–174, 2010.

[11] J. Liu, Z. Li, J. Tang, Y. Jiang, and H. Lu. Personalized geo-specific tag recommendation for photos on social websites. *IEEE Transactions on Multimedia*, 16(3):588–600, 2014.

[12] P. Lops, M. De Gemmis, G. Semeraro, C. Musto, and F. Narducci. Content-based and collaborative techniques for tag recommendation: an empirical evaluation. *Journal of Intelligent Information Systems*, 40(1):41–61, 2013.

[13] Y.-T. Lu, S.-I. Yu, T.-C. Chang, and J. Y.-j. Hsu. A content-based method to enhance tag recommendation. In *IJCAI*, volume 9, pages 2064–2069, 2009.

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

[15] H. Murfi and K. Obermayer. A two-level learning hierarchy of concept based keyword extraction for tag recommendations. *ECML PKDD discovery challenge*, pages 201–214, 2009.

[16] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *KDD*, pages 569–577, 2008.

[17] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256, 2009.

[18] D. Ramage, C. D. Manning, and S. Dumais. Partially labeled topic models for interpretable text mining. In *KDD*, pages 457–465, 2011.

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

[20] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, pages 81–90, 2010.

[21] A. K. Saha, R. K. Saha, and K. A. Schneider. A discriminative model approach for suggesting tags automatically for stack overflow questions. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 73–76. IEEE Press, 2013.

[22] P. Seitlinger, D. Kowald, C. Trattner, and T. Ley. Recommending tags with a model of human categorization. In *CIKM*, pages 2381–2386, 2013.

[23] B. Sigurbjörnsson and R. Van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW*, pages 327–336, 2008.

[24] Y. Song, L. Zhang, and C. L. Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1):4, 2011.

[25] S. Sood, S. Owsley, K. J. Hammond, and L. Birnbaum. Tagassist: Automatic tag suggestion for blog posts. In *ICWSM*, 2007.

[26] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys*, pages 43–50, 2008.

[27] H. Wang, X. Shi, and D.-Y. Yeung. Relational stacked denoising autoencoder for tag recommendation. In *AAAI*, pages 3052–3058, 2015.

[28] J. Wang, L. Hong, and B. D. Davison. Tag recommendation using keywords and association rules. In *KDD*, 2009.

[29] T. Wang, H. Wang, G. Yin, C. X. Ling, X. Li, and P. Zou. Tag recommendation for open source software. *Frontiers of Computer Science*, 8(1):69–82, 2014.

[30] X. Xia, D. Lo, X. Wang, and B. Zhou. Tag recommendation in software information sites. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 287–296. IEEE Press, 2013.

[31] M. Xu, R. Jin, and Z.-H. Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *NIPS*, pages 2301–2309, 2013.