User Embedding for Scholarly Microblog Recommendation

Yang Yu, Xiaojun Wan and Xinjie Zhou
Institute of Computer Science and Technology, The MOE Key Laboratory of Computational Linguistics, Peking University, Beijing 100871, China
{yu.yang, wanxiaojun, xinjiezhou}@pku.edu.cn

Abstract

Nowadays, many scholarly messages are posted on Chinese microblogs and more and more researchers tend to find scholarly information on microblogs. In order to exploit microblogging to benefit scientific research, we propose a scholarly microblog recommendation system in this study. It automatically collects and mines scholarly information from Chinese microblogs, and makes personalized recommendations to researchers. We propose two different neural network models which learn the vector representations for both users and microblog texts. Then the recommendation is accomplished based on the similarity between a user's vector and a microblog text's vector. We also build a dataset for this task. The two embedding models are evaluated on the dataset and show good results compared to several baselines.

1 Introduction

Online social networks such as microblogs have drawn growing attention in recent years, and more and more researchers are involved in microblogging websites. Besides expressing their own emotions and exchanging their life experiences just like other users, these researchers also write from time to time about their latest findings or recommend useful research resources on their microblogs, which may be insightful to other researchers in the same field. We call such microblog texts scholarly microblog texts. The volume of scholarly microblog texts is huge, which makes it time-consuming for a researcher to browse and find the ones that he or she is interested in.

In this study, we aim to build a personalized recommendation system for recommending scholarly microblogs. With such a system a researcher can easily obtain the scholarly microblogs he or she has interests in. The system first collects the latest scholarly microblogs by crawling from manually selected microblog users or by applying scholarly microblog classification methods, as introduced in (Yu and Wan, 2016). Second, the system models the relevance of each scholarly microblog to a researcher and make personalized recommendation. In this study, we focus on the second step of the system and aim to model the interest and preference of a researcher by embedding the researcher into a dense vector. We also embed each scholarly microblog into a dense vector, and thus the relevance of a scholarly microblog to a researcher can be estimated based on their vector representations.

In this paper, we propose two neural embedding algorithms for learning the vector representations for both users (researchers) and microblog texts. By extending the paragraph vector representation method proposed by (Le and Mikolov, 2014), the vector representations are jointly learned in a single framework. By modeling the user preferences into the same vector space with the words and texts, we can obtain the similarity between them in a straightforward way, and use this relevance for microblog recommendation. We build a real evaluation dataset from Sina Weibo. Evaluation results on the dataset show the efficacy of our proposed methods.

2 Related Work

There have been a few previous studies focusing on microblog recommendation. Chen et al. (2012) proposed a collaborative ranking model. Their approach takes advantage of collaborative filtering based recommendation by collecting preference information from many users. Their approach takes into account the content of the tweet, user's social relations and certain other explicitly defined features. Ma et al. (2011) generated recommendations by adding additional social regularization terms in MF to constrain the user latent feature vectors to be similar to his or her friends' average latent features. Bhattacharya et al. (2014)

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 449–453,
Berlin, Germany, August 7-12, 2016.

proposed a method benefiting from knowing the user's topics of interest, inferring the topics of interest for an individual user. Their idea is to infer them from the topical expertise of the users whom the user follows. Khater and Elmongu (2015) proposed a dynamic personalized tweet recommendation system capturing the user's interests, which change over the time. Their system shows the messages that correspond to such dynamic interests. Kuang et al. (2016) considered three major aspects in their proposed tweet recommending model, including the popularity of a tweet itself, the intimacy between the user and the tweet publisher, and the interest fields of the user. They also divided the users into three types by analyzing their behaviors, using different weights for the three aspects when recommending tweets for different types of users.

Most of the above studies make use of the relationships between users, while in this study, we focus on leveraging only the microblog texts for addressing the task.

3 Our Approach

3.1 Task Definition

We denote a set of users by $u = \{u_1, u_2, \ldots, u_m\}$, and a set of microblog texts by $d = \{d_1, d_2, \ldots, d_n\}$. We assume that a user tweeting, retweeting or commenting on a microblog text reflects that the user is interested in that microblog. Given $u_t \in u$, we denote the set of microblogs that u_t is interested in by $d(u_t)$. In our task, the entire sets of d and u are given, while given a user $u_t \in u$, only a subset of $d(u_t)$ is known. This subset is used as the training set, denoted as $\tilde{d}(u_t)$. Our task aims to retrieve a subset d' of d, that d' is as similar to $d(u_t) - \tilde{d}(u_t)$ as possible.

In this section, we introduce one baseline method and then propose two different neural network methods for user and microblog embedding. The baseline averages the vector representation of microblog texts into a user vector representation. Our proposed two methods learn user vector representations jointly with word and text vectors, either indirectly or directly from word vectors.

3.2 Paragraph Vector

As our methods are mainly based on the Paragraph Vector model proposed by (Le and Mikolov, 2014), we start by introducing this framework first.

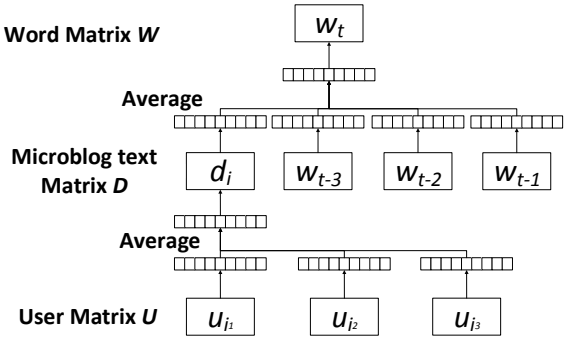

Figure 1. The proposed User2Vec#1 framework for learning user vector representation. In this framework, the word vectors do not directly contribute to the user vectors.

Paragraph Vector is an unsupervised framework that learns continuous distributed vector representations for pieces of texts. In this approach, every paragraph is mapped to a unique vector, represented by a column in matrix D and every word is also mapped to a unique vector, represented by a column in matrix W. This approach is similar to the Word2Vec approach proposed in (Mikolov et al., 2013), except that a paragraph token is added to the paragraph and is treated as a special word. The paragraph vector is asked to contribute to the prediction work in addition to the word vectors in the context of the word to be predicted. The paragraph vector and word vectors are averaged to predict the next word in a context.

Formally speaking, given a paragraph $\{d_i, w_1, w_2, \ldots, w_T\}$ with d_i as the paragraph token, k as the window size, the Paragraph Vector model applies hierarchical softmax to maximize the average log probability

$$\frac{1}{T}\sum_t \log p(w_t \mid d_i, w_{t-k}, ..., w_{t+k})$$

3.3 Averaging Microblog Text Vectors as User Vector

An intuitive baseline approach to map a microblog user into a vector space is to build such representation from the vector representations of the microblogs he or she likes.

We treat microblog texts as paragraphs, and then apply the Paragraph Vector model introduced in Section 3.2 to learn vector representations of the microblog texts. After learning all vector representations of microblog texts, for each user, we average all vectors of microblog

texts he or she likes in the training set as the user vector.

3.4 Learning User Vectors Indirectly From Word Vectors

Besides the above-mentioned baseline approach we further consider to jointly learn the vectors of users and microblog texts. In this framework, every user is mapped to a vector represented in a column in matrix U, in addition to the microblog text matrix D and the word matrix W. Given a microblog text $\{d_i, w_1, w_2, \ldots, w_T\}$, besides predicting words in the microblog texts using the microblog token d_i and words in the sliding window, we also try to predict d_i using the users related to it. Denoting the set of all users related to d_i in the training set as $\tilde{u}(d_i) = \{u_{i_1}, u_{i_2} ..., u_{i_h}\}$, we maximize the average log probability

$$\frac{1}{T}\sum_t [\log p(w_t \mid d_i, w_{t-k}, ..., w_{t+k}) + \log p(d_i \mid u_{i_1}, ..., u_{i_h})]$$

The structure of this framework is shown in Figure 1. We name this framework User2Vec#1.

3.5 Learning User Vectors Directly From Word Vectors

In the above framework, the user vectors are learned only from microblog text vectors, not directly from word vectors. Another framework we proposed for learning user vector representation is to put user vectors and microblog vectors in the same layer. Unlike User2Vec#1, we do not use user vectors to predict microblog text vector. Instead, we directly add user vectors into the input layer of word vector prediction task, along with the microblog text vector.

In this framework, the average log probability we want to maximize is

$$\frac{1}{T}(\sum_t \log p(w_t \mid d_i, w_{t-k}, ..., w_{t+k}, u_{i_1}, ..., u_{i_h})$$

In practical tasks, we modify the dataset by copying each microblog once for each user in $\tilde{u}(d_i)$, and make each copied microblog text only relate to one user. All copies of the same microblog text share a same vector representation.

The structure of the framework is shown in Figure 2. We name this framework User2Vec#2.

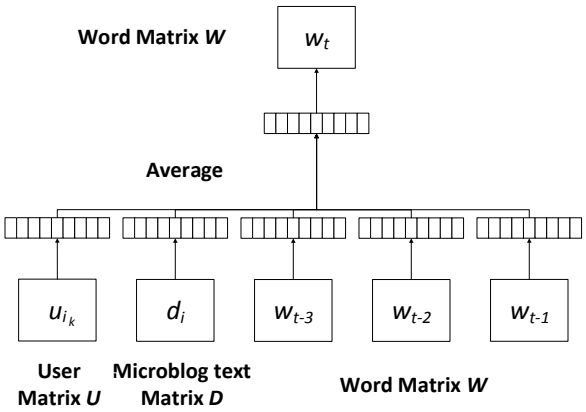

Figure 2. The proposed User2Vec#2 framework for learning user vector representation. In this framework, the word vectors contribute directly to the user vectors, along with the microblog text vectors.

3.6 Recommending Microblogs

When recommending microblogs, given a microblog d_j and a user u_k, we compute the cosine distance between their vector representations, and use the cosine distance to determine whether d_j should be recommended to u_k or not.

4 Evaluation

4.1 Data Preparation

To evaluate our proposed user embedding methods in a scholarly microblog recommending system, we built a dataset by crawling from the website Machine Learning Daily[1].

The Machine Learning Daily is a Chinese website which focuses on collecting and labeling scholarly microblogs related to machine learning, natural language processing, information retrieval and data mining on Sina Weibo. These microblog texts were collected by a combination of manual and automatic methods, and each microblog text is annotated with multiple tags by experts, yielding an excellent dataset for our experiment. The microblog texts in our dataset can be written in a mixture of both Chinese and English. We removed stop words from the raw texts, leaving 16,797 words in our corpus. The texts were then segmented with the Jieba Chinese text segmentation tool[2].

[1] http://ml.memect.com/
[2] https://github.com/fxsjy/jieba

	k=10			*k*=20			*k*=50			*k*=100		
	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR	Precision	Recall	MRR
Bag-of-Words	0.5036	0.0504	0.0153	0.4917	0.0983	0.0185	0.4461	0.2231	0.0223	0.3204	0.3204	0.0246
SVM on BoW	0.5774	0.0577	0.0172	0.5662	0.1132	0.0212	0.5122	0.2561	0.0256	0.3675	0.3675	0.0282
Average Embedding	0.5963	0.0596	0.0183	0.5824	0.1165	0.0219	0.5266	0.2633	0.0264	0.3793	0.3793	0.0291
User2Vec#1	0.6246	0.0625	0.0189	0.6055	0.1211	0.0228	0.5511	0.2756	0.0275	0.3953	0.3953	0.0304
User2Vec#2	**0.6652**	**0.0665**	**0.0201**	**0.6498**	**0.1300**	**0.0244**	**0.5883**	**0.2942**	**0.0295**	**0.4231**	**0.4231**	**0.0325**

Table 1. Overview of results.

After crawling the microblogs from the Machine Learning Daily, we used Sina Weibo API to retrieve the list of users who retweeted or commented on those microblogs. These retweeting and commenting actions indicated that those users have interests in the microblogs they retweeted or commented, and such microblogs were considered the gold-standard (positive) microblogs for the users in the recommendation system. Then we filtered out the users who have less than two hundred positive samples to avoid the data sparseness problem. This left us with 711 users and 10,620 microblog texts in our corpus. Each user was associated with 282.3 positive microblogs on average.

4.2 Evaluation Setup

Because there is no API that can directly grant us the access to the follower and followee list for each user without authorization on Sina Weibo, when evaluating the effectiveness of our methods, we randomly choose one hundred positive samples and another four hundred negative samples randomly selected from the crawled microblogs, to simulate the timeline of a user, and use this simulated timeline as the test dataset. The remaining positive samples are used for training.

We adopt two additional baselines: Bag-of-Words and SVM on Bag-of-Words. For the Bag-of-Words baseline, we use the Bag-of-Words vector of each microblog text as the microblog text vector, and average them to obtain user vectors. For the SVM on Bag-of-Words baseline, we randomly choose the same amount of negative samples as that of positive samples for training. We use the Bag-of-Words vector of each microblog text as the features, and run the SVM algorithm implemented in LibSVM [3] once for every user. Note that the Average Embedding method introduced in Section 3.3 is considered a strong baseline for comparison.

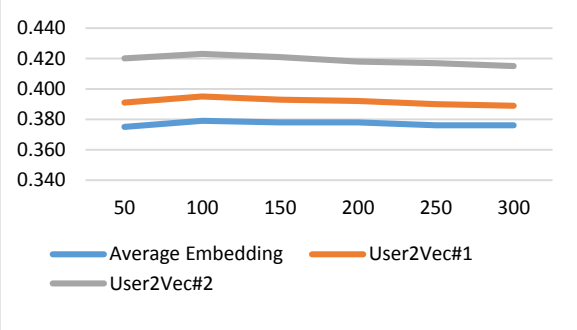

Figure 3. Precision/Recall@*k*=100 w.r.t. vector dimension.

For each method and each user, we sort the microblog texts according to their similarity with the user and select the top *k* microblog texts as recommendation results, where *k* varies from 10 to 100.

Besides precision and recall values, we also compute mean reciprocal rank (MRR) to measure the recommendation results in our experiments, which is the average of the multiplicative inverse of the rank of the positive samples in the output of the recommending system, and then averaged again across all users. Note that when *k* is set to 100, the precision and recall value will be equal to each other.

4.3 Evaluation Results

The comparison results with respect to different *k* are shown in Table 1. As we can see, the two proposed joint learning methods outperform the simple average embedding method and the two other baselines, indicating the effectiveness of the proposed methods. Moreover, User2Vec#2 yields better results than User2Vec#1.We believe this is because in User2Vec#2, the word vectors have a direct contribution to the user vectors, which improves the learning effect of the user

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

vectors learnt in the framework. Furthermore, the precision/recall scores of the embedding methods (k=100) with respect to different vector dimensions are shown in Figure 3. We can see that the dimension size has little impact on the recommendation performance, and our proposed two methods always outperform the strong baseline.

5 Conclusion

In this paper, we proposed two neural embedding methods for learning the vector representations for both the users and the microblog texts. We tested their performance by applying them to recommending scholarly microblogs. In future work, we will investigate leveraging user relationships and temporal information to further improve the recommendation performance.

Acknowledgments

The work was supported by National Natural Science Foundation of China (61331011), National Hi-Tech Research and Development Program (863 Program) of China (2015AA015403) and IBM Global Faculty Award Program. We thank the anonymous reviewers and mentor for their helpful comments. Xiaojun Wan is the corresponding author.

References

Michal Barla. 2011. Towards social-based user modeling and personalization. *Information Sciences and Technologies Bulletin of the ACM Slovakia*, 3(1).

Parantapa Bhattacharya, Muhammad Bilal Zafar, Niloy Ganguly, Saptarshi Ghosh, and Krishna P. Gummadi. 2014. Inferring user interests in the twitter social network. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM.

Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. 2012. Collaborative personalized tweet recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Shaymaa Khater and Hicham G. Elmongui. 2015. Tweets You Like: Personalized Tweets Recommendation based on Dynamic Users Interests. In *2014 ASE Conference.*

Li Kuang, Xiang Tang, Meiqi Yu, Yujian Huang and Kehua Guo. 2016. A comprehensive ranking model for tweets big data in online social network. *EURASIP Journal on Wireless Communications and Networking, 2016(1).*

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053.*

Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining.* ACM.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu and Xiaofang Zhou. 2015. Dynamic user modeling in social media systems. *ACM Transactions on Information Systems (TOIS)*, 33(3).

Jianjun Yu, Yi Shen and Zhenglu Yang. 2014. Topic-STG: Extending the session-based temporal graph approach for personalized tweet recommendation. In *Proceedings of the companion publication of the 23rd international conference on World Wide Web companion.* International World Wide Web Conferences Steering Committee.

Yang Yu and Xiaojun Wan. 2016. MicroScholar: Mining Scholarly Information from Chinese Microblogs. In *Thirtieth AAAI Conference on Artificial Intelligence*.