UIR-SIST System for Identifying User Profile by Neural Networks on CSDN Dataset

Junru Lu[1], Le Chen[1], Kongming Meng[2], Fengyi Wang[3], Jun Xiang[1], Nuo Chen[1], Kaimin Zhou[2], Binyang Li[1]*

[1] School of Information Science and Technology, University of International Relations, Beijing, China
[2] Deep Brain Co. Ltd. Shanghai, China
[3] University of Chinese Academy of Sciences, Beijing China

ABSTRACT

As the popularity of social media, user profiling nowadays becomes a very hot research topic and an emerging application. This paper presents our system named UIR-SIST developed by ELP team for SMP CUP 2017 user profiling evaluation. UIR-SIST targets on three subtasks, including keywords extraction from blog, user interests labeling, and user growth prediction. We firstly propose to capture three aspects information to extract keywords from blog, including blog itself, blogs belonging to the same topic, and the blogs published by the same user. Then a unified neural network model is constructed for user interests tagging. Finally, we adopt a stacking model for predicting user growth trending value. Based on SMP CUP 2017's metrics, our model runs got final scores of 0.563, 0.378 and 0.751 on 3 tasks respectively.

1. INTRODUCTION

Recently, social media becomes an important platform for online users to generate and spread valuable information. These user-generated contents in social media cannot only directly help people to find answers or make decisions, but also become strong evidences to investigate some aspects of users. For instance, Chinese Software Developer Network (CSDN) is considered as a biggest forum in China for software engineers to share technical information, engineering experience, relevant question&answering, and advertising et al. Meanwhile, the user-generated contents on CSDN also provides an opportunity to display users' special interests in the software developing aspect of themselves, such as their past interests, current focus, and growth trending values, even though their user profiles are incomplete or even missing. Moreover, accompany with the user-generated contents, user behaviors also contain useful information about user profile, such as "following", "reply", and "private letter" behaviors, through which the friendship network is constructed to indicate user gender [1-3], age [4], political polarity [5,6], or profession [6]. Therefore, much attention on the research of user profiling has been attracted from both academic and industrial areas.

SMP CUP has been held for user profiling evaluation since 2016, which aimed at encouraging participators to effectively model user profile based on given dataset. This year comes the second edition, i.e. SMP CUP 2017, and three subtasks are established based on CSDN blogs[21]:

(1) Extract three keywords from each given blog
(2) Tag user interests with three labels
(3) Predict user growth trending value

ELP team from School of Information Science and Technology, University of International Relations (UIR SIST) participated all the subtasks in SMP CUP 2017. This paper describes the framework of UIR SIST for SMP CUP 2017. We firstly propose to capture three aspects information to extract keywords from blog, including blog itself, blogs belonging to the same topic, and the blogs published by the same user. Then a unified neural network model is constructed for Task 2. The model is based on multi-scale convolutional neural networks whose aim is to capture local and global information to modeling user. Finally, we adopt stacking model for predicting user growth trending value. According to SMP CUP 2017's metrics, our model runs got final scores of 0.563, 0.378 and 0.751 on 3 tasks respectively.

This paper is organized as follows. Section 2 introduce the SMP CUP 2017 evaluation in details. Section describe the framework of our system. We then present the evaluation results in Section 4. Finally, Section 5 concludes the paper.

2. Evaluation Overview

This year comes the second edition of SMP CUP for user profiling, and three tasks are established based on CSDN dataset, including keywords extraction from CSDN blogs, user interests tagging, and user growth

* Corresponding author: Binyang Li, byli@uir.edu.cn.

trending value prediction. In this section, we will introduce the dataset, and then describe the tasks as well as the evaluation metrics in detail.

2.1 Dataset

The dataset used in SMP CUP 2017 is provided by CSDN, which is the biggest online Chinese IT community. The CSDN dataset consists of all the user-generated contents and the behavior information from 157,427 users during 2015, which can be further divided into three parts:

(1) 1,000,000 pieces of users' blogs, involving blog ID, blog title and the corresponding contents;
(2) 6 types of user behavior information, including Post, Browse, Comment, Vote up, Vote down, and Favorite, and the corresponding date and time information;
(3) Relationship between users, which specifically refers to the records of Follow and Private letters.

More detailed information about the size and type of the CSDN dataset are shown in Table 1.

Table 1. Statistics of the Evaluation Dataset

Attribute		**Content**	**Size**	**Format**
Blogs		Users' blogs	1,000,000	D0802938/Title/Content
Behavior	**Post**	record of posting blogs	1,000,000	U0024827/D0874760/2015-02-05 18:05:49.0
	Browse	record of browsing blogs	3,536,444	U0143891/D0122539/20150919 09:48:07
	Comment	record of commenting on blogs	182,273	U0075737/D0383611/2015-10-30 11:18:32.0
	Vote up	record of voting blogs up	95,668	U0111639/D0627490/2015-02-21 12:21:12
	Vote down	record of voting blogs down	9,326	U0019111/D0582423/2015-11-23 22:54:48
	Favorite	record of favoriting blogs	10,4723	U0014911/D0552113/2015-06-07 07:05:05
Relationships	**Follow**	record of follow relationships	667,037	U0124114/U0020107
	Letter	record of sending private letters	46, 572	U0079109/U0055181/2015-12-24 01:09:38.0

More detailedly, Table 2 illustrates an example.

Table 2. Sample of CSDN dataset.

Attribute	**Data sample**
User ID	U00296783
Blog ID	D00034623
Blog Content	Title:[转]使用TextRank算法为文本生成关键字和摘要; Content: TextRank算法基于PageRank...
Blog Keywords	Keyword1: TextRank; Keyword2: PageRank; Keyword3: 摘要
Interest Tags	Tag1: 大数据; Tag2: 数据挖掘; Tag3: 机器学习
Post	U00296783 / D00034623 / 20160408 12:35:49
Browse	D09983742 / 20160410 08:30:40
Comment	D09983742 / 20160410 08:49:02
Vote up	D00234899 / 20160410 09:40:24
Vote down	D00098183 / 20160501 15:11:00
Letter	U00296783 / U02748273 / 20160501 15:30:36
Favorite	D00234899 / 20160410 09:40:44
Follow	U00296783 / U02666623 / 20161119 10:30:44
Growth Value	0.0367

2.2 Tasks

In SMP CUP 2017, 3 specific subtasks relevant to user profiling are set up based on the above CSDN dataset.

Task 1: It is required to extract 3 keywords from each document that can well represent the topic or the main content of the document.

Task 2: It is required to generate 3 labels to describe users' interests, where the labels are chosen from a given candidate set (42 in total).

Task 3: It is required to predict each user's growth trending of the next six months according to his/her behavior of the past year, including the texts, the relationships, and the interactions with other users. The growth trending needs to be scaled into [0, 1], where 0 presents the drop-out of a user.

2.3 Metrics

To assess the effectiveness of the above tasks, the following evaluation metrics are designed for each individual subtask.

$Score_1$ is defined to calculate the overlapping ratio between the extracted keywords and the standard answers, which can be computed as follows:

$$Score_1 = \frac{1}{N}\sum_{i=1}^{N}\frac{|K_i \cap K_i^*|}{|K_i|}$$

where N is the size of the valid set or the test set, K_i is the extracted keywords set from document i, and K_i^* is the standard keywords of document i. Note that it is defined that $\left|K_i\right| = 3$ and $\left|K_i^*\right| = 5$.

$Score_2$ denotes the overlapping ratio of model tagging and answers, which can be expressed by the following equation:

$$Score_2 = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathrm{T}_i \cap \mathrm{T}_i^*|}{|T_i|}$$

where T_i is the automatically generated tag set of user i, and $\mathrm{T}_i^*$is the standard tags of user i. It is also defined that $|\mathrm{T}_i|$=3 and $|\mathrm{T}_i^*|$=3.

$Score_3$ is calculated by relative error between the predicted growth trending value and the real growth value of users, which can be expressed by the following equation:

$$Score_3 = 1 - \frac{1}{N}\sum_{I=1}^{N}\begin{cases} 0, & v_i = 0, v_i^* = 0 \\ \left|v_i - v_i^*\right| / \max\left(v_i, v_i^*\right), & otherwise \end{cases}$$

where v_i is the predicted growth trending value of user i, and v_i^* is the real growth value of user i.

The overall can be computed as $Score_{all} = Score_1 + Score_2 + Score_3$.

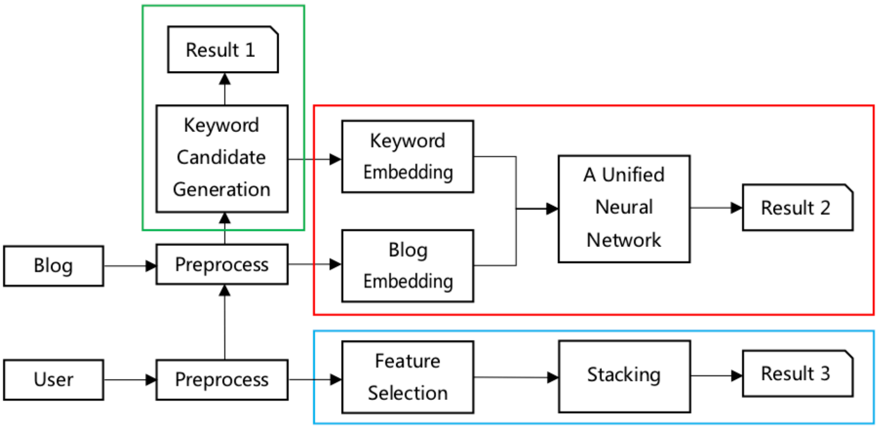


Figure 1. System Architecture.

3. System Overview

The overall architecture of UIR-SIST is described in Figure 1. UIR-SIST system is comprised of 4 modules:

(1) Preprocess module: reads all blogs of training set and test set. It performs word segmentation, POS tagging, named entity recognition, and semantic role labeling[19];

(2) Keyword extraction module: extract 3 keywords to represent the main content of the blog, which can be captured from three aspect to generate the candidate keywords set, including the blog itself, the blogs published from the same user, and the blogs belonging to the same topic, as shown in green part;

(3) User interests tagging module: construct a combined neural networks with user's content embedding and keyword&user tag embedding[22] for user interests tagging, as shown in red part;

(4) User growth trending value prediction module: implement users' interaction information and the behavior features into a supervised learning model for growth trending prediction.

We will then describe each module in details.

3.1 Keywords Extraction

The objective of subtask-1 is to extract 3 keywords from each blog that can represent the main content of the blog. In our opinion, the main content can be captured from the following three aspects, the blog itself, the blogs published from the same user, and the blogs belonging to the same topic. Based on this assumption, we adopt three different models that can capture each aspect content to generate a candidate keywords set. And then 3 keywords are extracted from the candidate set by using some rules.

We firstly adopted the classic *tf-idf* weighting scheme to reflect the content of the blog itself. Then we rank the keywords based on the *tf-idf* score, and select the top 100 keywords from the result to form the candidate keyword set.

To account for the blogs belonging to the same user, we adopt TextRank approach[16] to cluster their blogs together. Meanwhile, all the keywords will be weighed during this processing. We finally select the top 300 keywords.

Moreover, we also utilize topic information to extract the keywords. Since 42 categories of tags are given in Task 2, we assume that these 42 topics are extracted from all the blogs. Therefore, we use Latent Dirichlet Allocation (LDA) model[7] to extract top 100 keywords for each category from 1,000,000 blogs, and thus obtained the interspecific distribution information of these 4200 subject keywords.

In summary, we take three aspects to reflect the blog content and get three independent candidate keywords sets, which are extracted through *tf-idf* model, Textrank model, and LDA model. After that, we only save the intersection set of three candidates. In our Training Set of Task 1, there are about 5,000 keywords provided, which are collected after extraction and deduplication by our team.

As we know, the classic *tf-idf* model suffers from the simply presuppose that the rarer a word is in corpus, the more important it is, and the greater its contribution is to the meaning of the text. However, when referring to a group of articles, which mainly use same keywords and describe some similar concepts, the calculation results will have many errors. This is also the reason we use *tf-idf* in the short single blog, and use the TextRank model in the larger blog collection belonging to the same user, because the larger corpus will amplify the *tf-idf*'s errors.

In addition, in order to enhance its cross-topic analysis ability, we borrow the idea of 2016 CCF Evaluation[18], and implement the improvements on the results of traditional TF-IDF calculation, and get the result *S-TFIDF*(*w*) by using the following equation:

$$\mathrm{S-TFIDF(w) = TFIDF(w)}*(\frac{1}{C_w}-\frac{1}{42})$$

where C_w is the number of the times of word w appearing in 42 categories.

3.2 User Interests Tagging

The objective of this subtask is to tag the user interests with 3 labels from 42 given ones. We model this subtask with neural networks, and the model structure is shown in Figure 2. Each blog is represented by a blog embedding[8, 22] through convolution and max-pooling layers. Then we obtain a user's content embedding from weighted sum of all blog embedding of the user. The weight value of each blog embedding is counted by self-attention mechanism. Content embedding and keyword embedding are concatenated as user embedding, and finally fed to output layer.

In our system, a CNN (convolutional neural network) model is constructed for blog representation instead of RNN, since more global information will be captured for indicating the user interests and the time efficiency will also be enhanced. It is widely acknowledged that multi-scale convolutional neural networks[12] have implemented due to its outstanding achievement on computer vision[13], and TextCNNs is designed by arraying word embedding vertically, has also shown quite high effectiveness for NLP tasks[14].

In our CNNs model, we view a blog as a sequence of words $x = [x_1, x_1, \cdots, x_1]$ where each one is represented by its word embedding vector, and returns a feature matrix S of the blog. The narrow convolution layer attached after the matrix is based on a kernel $W \in R^{kd}$ of width k, a nonlinear function f and a bias variable b:

$$h_i = f\left(W_{xi:i+k-1} + b\right)$$

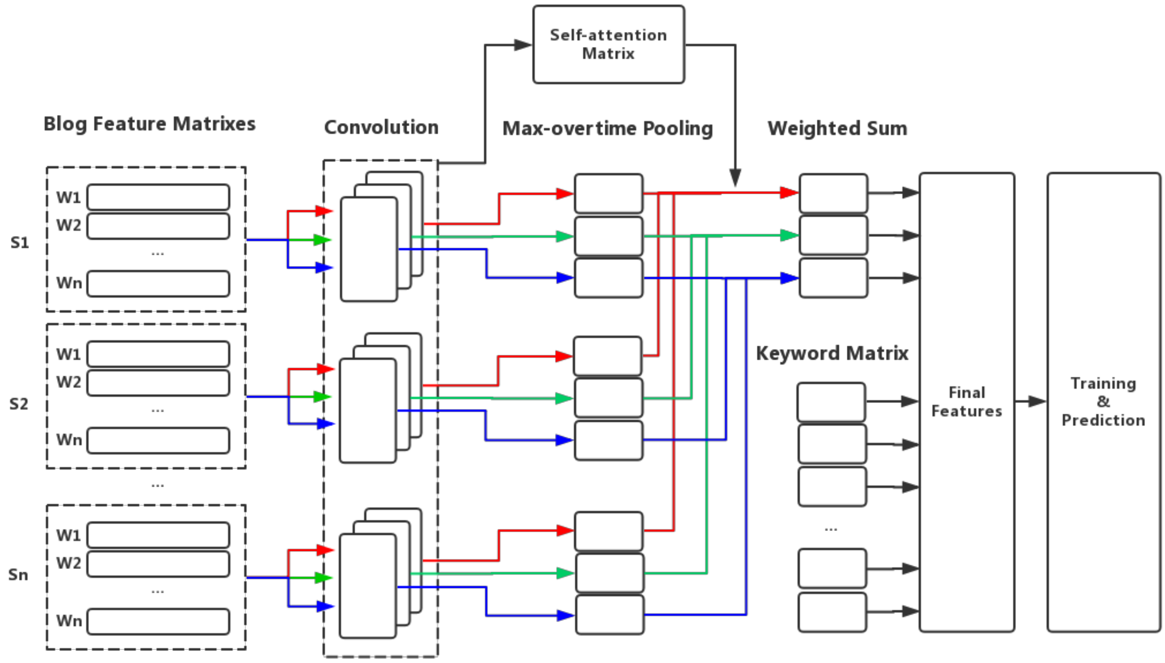


Figure 2. Framework of CNNs model based on Weighted-Blog-Embeddings in Task 2.

where $xi{:}j$ refers, specifically, to the concatenation of the sequence of words' vectors from position i to position j. In this subtask, we use several kernel sizes to obtain multiple local contextual feature maps in convolution layer, and then apply the max-overtime pooling[15] to extract some most important features.

The output of that is the low-dimensional dense and quantified representation of each single blog. After that, each user has all their relevant blogs computable. We can simply average their blogs' vectors to get the content embedding $c(u)$ for any individual user:

$$c(u) = \frac{1}{T}\sum_{i=1}^{T} s_i$$

where T is the total number of a user's related blogs.

However, different sources of blogs imply the extent of user's interest on different topics. For example, a blog posted by a user may be generated from writing by himself, reposting from other users, or sharing from another platform. It is naturally that we may want to pay different degree of attention on these blogs when we tend to infer the user's interests. Thus, a self-attention mechanism is introduced, which automatically assigns different weight value for each blog of a user after training. The user context representation is given by weighted summation of all blogs' vectors:

$$\alpha = \frac{\exp(e_i)}{\sum_{j=1}^{T} e_j}$$

$$e_i = v^T \tanh(Ws_i + Uh_i)$$

$$c(u) = \sum_{i=1}^{T} \alpha_i h_i$$

where α_i is the weight of i-th blog, s_i is the one-hot source representation vector of the blog, $v \in R^{n'}$, $W \in R^{n' \times m}$, $U \in R^{n' \times n}$, $s_i \in R^m$, $h_i \in R^n$, and m is the number of all source platforms.

When we finish the user's context representation, the keyword matrix of all blog's keywords extracted by model in subtask 1 will be concatenated. The final features is the output of above whole feature engineering.

Afterwards, an ANN layer trains the user embeddings from the train set and predict probability distribution of users' interests among 42 tags in valid and test set according to their embeddings.

3.3. User Growth Trending Prediction

According to the description of Task 3, we consider that the growth trending value can be estimated as the degree of activeness. Therefore, our basic idea is to implement users' interaction information and the behavior statistical features into a supervised learning model. The procedure of Task 3 is demonstrated by Figure 3.

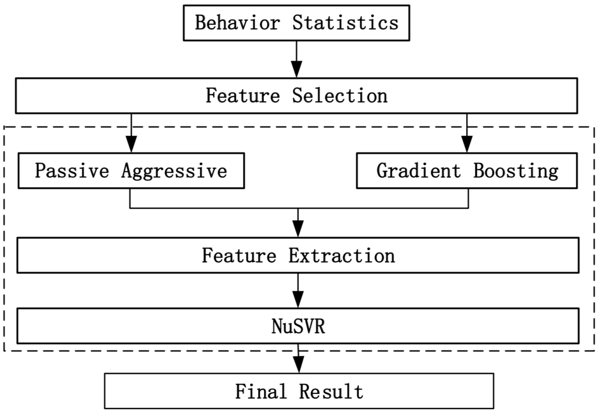
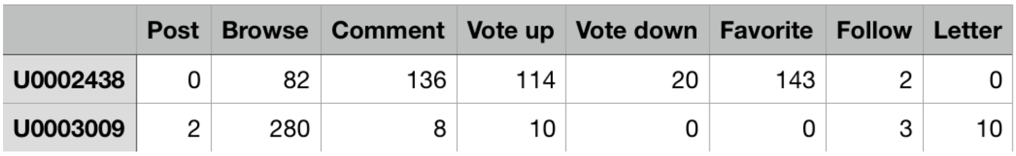


Figure 3. Framework of Stacking model in Task 3.

On the whole, we use a stacking framework to enhance the accuracy of final prediction[17]. After the basic behavior statistics analysis, the original feature are selected as input into the stacking model. Then, the stacking model is divided into two layer, the base layer and the stacking layer. In the base layer, we choose Passive Aggressive Regressor[10] and Gradient Boosting Regressor[11] as the group of basic regressors due to their excellent performance. In the stacking layer, we still use SVM model, especially, the NuSVR model, which can control its error rate. Finally, we get the final result of user growth trending value.

3.3.1 Original Feature Selection

Figure 4 illustrates an example of the daily statistics of user behaviors, including post, browse, comment, vote up, vote down, favorite, follow, and private letter. To predict the user growth trending value, it is intuitively that the dynamic changes of behaviors along the time line is more useful. To avoid of data sparse, we adopt users' monthly statistics of user behaviors rather than daily statistics.



	Post	Browse	Comment	Vote up	Vote down	Favorite	Follow	Letter
U0002438	0	82	136	114	20	143	2	0
U0003009	2	280	8	10	0	0	3	10

Figure 4. Example of daily statistics of user behavior.

Then we use correlation analysis to exclude the "vote down" behavior because of its negative contribution to model prediction. After that, through feature selection, we use the average, Log calculation and growth rate of original data to obtain features for stacking model.

$$\mathrm{LOG}\left(\mathrm{d}\right) = \log(\mathrm{d} + 1)$$

$$\mathrm{GR}\left(d_t\right) = \frac{d_{t+1} - d_t}{d_t + 1}$$

where $LOG(d)$ represents the calculation result of data d after adjustment, and $GR(d_t)$ represents the calculation result of growth value from data d_t in month t to data d_{t+1} in month $t+1$.

3.3.2 PAR/GDR-NuSVR-Stacking Model(PGNS)

Once we have gotten monthly statistics and derivative features as described above, the combination of them will be sent as inputs into Passive Aggressive Regressor and Gradient Boosting Regressor independently. By averaging the predictions of those two base models, a new feature will be created and input into the stacking model NuSVR. Because of the randomness of base models, a self-check mechanism that we have adopt 10-fold cross validation.

If the trained model gets a score higher than the threshold S^* under given scoring rules, we will enter the corresponding features of Valid Set or Test Set into the model to get a prediction, which will be saved into *Candidate Set*. On the contrary, if the trained model gets a 10-fold cross validation score that is lower than S^*, the model will be discarded and the program will return to the training session shown in the dotted box for a new round of training.

In order to reduce the error of single round of training, we set at least R^* rounds for training and add all predictions that get higher scores than S^* to the candidate set. According to experience, the ratio of the size of *Candidate Set* to R^* is about 0.45. When all rounds of trainings are completed, all predictions in the *Candidate Set* will be calculated to generate an average prediction as the final result.

4. Evaluation

In this section, we will report our evaluation results in SMP CUP 2017 based on the given dataset as well as the parameter setting. In our mode, we firstly trained the word embedding with the dimensions of 300. For the CNN model, we set the sequence_length as 300, the num_classes as 42, dropout as 0.5, number of filters as 128, and the filter_sizes as 3, 4, 5, respectively.

Table 1 shows the comparison result of our proposed approach for Task 1 with the consideration of different aspects to capture the blog content. From the results, we can see that the best results were achieved when all the aspects information were used for capturing blog content.

Table 3. Comparison of incorporating different aspects on Task 1.

Approach	Results
BI: Blog Itself	0.505
ST: Same Topic	0.371
SU: Same User	0.436
BI+ST+SU	**0.563**

Besides, we also test performance of our combined neural network with different embedding inputs. Note that to get the result of individual embedding, we trained a new CNN model for blog embedding, and we compute the similarity between blog content and keyword in embedding representation. The experimental results are shown in Table 4.

Table 4. Comparison of different aspects on Task 2.

Approach	Details
Blog Embedding	0.301
Keywords Embedding	0.245
All	**0.378**

Finally, Table 6 display the overall performance of our UIR-SIST system's best run in SMP CUP 2017.

Table 5. Performance of UIR-SIST system in SMP CUP 2017.

	Task 1	Task 2	Task 3	Total
Train Set(10 Fold)	0.61	0.39	0.765	1.765
Valid Set	0.56	0.39	0.73	1.680
Test Set	0.563	0.378	0.751	1.692

5. Conclusions

In this paper, we present a framework for SMP CUP 2017 user profiling task. We firstly propose to capture three aspects information to extract keywords from blog, including blog itself, blogs belonging to the same topic, and the blogs published by the same user. Then a unified neural network model is constructed for Task

2. The model is based on multi-scale convolutional neural networks whose aim is to capture local and global information to modeling user. Finally, we adopt stacking model for predicting user growth trending value. According to SMP CUP 2017's metrics, our model runs got final scores of 0.563, 0.378 and 0.751 on 3 tasks respectively.

6. Future Research Directions

First of all, the use of the date sets can be improved. For example, when we were doing subtask-1, we focused more attention on the topics. However, the beginning and the end is also important. We should distribute weight to these parts of the blogs.

Secondly, 42 user interests tags are not balanced. They are hierarchical. Some tags is likely to be the subtags of another tag. We can add some tags into the tag space in order to form a complete system. Then the tags can be sorted more orderly.

Finally, we can consider more relationships between three subtasks. There are some connections between users and blogs. That's why the result of subtask-1 is helpful for processing the subtask-2. And subtask-3 has indicated that the analysis of time series should be taken into account. During the subtask-2, we only use the users' behavior on the blogs. Timing about their behavior fell into neglect. The weight value of each blog embedding can be adjusted according to their timing. Because the later a user focus on a blog, the blog is more likely to represent his latest interest.

Reference

1. Ciot, M., Sonderegger, M., Ruths, D.: Gender inference of twitter users in nonEnglish contexts. In: Proceedings of EMNLP, pp. 18–21 (2013) 2. Wendy, L., Derek, R.: What's in a name? Using first names as features for gender inference in twitter. In: AAAI Spring Symposium Series (2013)
2. Liu, W., Zamal, F.A., Ruths, D.: Using social media to infer gender composition of commuter populations. In: Proceedings of the International Conference on Weblogs and Social Media (2102)
3. Rao, D., Yarowsky, D.: Detecting latent user properties in social media. In: Proceedings of the NIPS MLSN Workshop (2010)
4. Pennacchiotti, M., Popescu, A.M.: A machine learning approach to twitter user classification. In: Proceedings of ICWSM (2011)
5. Conover, M.D., Ratkiewicz, J., Francisco, M., et al.: Political polarization on twitter. In: Proceedings of ICWSM (2011)
6. Tu, C., Liu, Z., Sun, M.: PRISM: Profession Identification in Social Media with personal information and community structure. In: Proceedings of Social Media Processing (2015)
7. Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3:993-1022.
8. Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.
9. Wolpert D H. Original Contribution: Stacked generalization[J]. Neural Netw, 1992.
10. Crammer K, Dekel O, Keshet J, et al. Online Passive-Aggressive Algorithms[J]. Journal of Machine Learning Research, 2006, 7(3):551-585.
11. Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001, 29(5): 1189-1232.
12. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a backpropagation network. In: Proceedings of NIPS (1989)
13. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Proceedings of NIPS (2012)
14. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
15. Collobert, R., Weston, J., Bottou, L., et al.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12(8), 2493–2537 (2011)
16. Mihalcea, R. and Tarau, P., 2004. Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing.
17. Friedman J. Stochastic Gradient Boosting[C]// 1999:367--378.
18. https://github.com/coderSkyChen/2016CCF_BDCI_Sougou
19. https://github.com/fxsjy/jieba
20. https://github.com/LuJunru/SMPCUP2017_ELP
21. https://biendata.com/competition/smpcup2017/
22. https://github.com/RaRe-Technologies/gensim