Identifying Users' Gender via Social Representations

Tieyun Qian[1], Peisong Zhu[1], Xuhui Li[2*], Dewang Sun[2]
[1]State Key Laboratory of Software Engineering, Wuhan University, Hubei, China
[2]School of Information Management, Wuhan University, Hubei, China
[*]Contact author

Abstract
Gender prediction has evoked great research interests due to its potential applications like targeted advertisement and personalized search. Most of existing studies rely on the content texts. However, the text information is hard to access. This makes it difficult to extract text features.
In this paper, we propose a novel framework which only involves the users' ids for gender prediction. The key idea is to represent users in the embedding connection space. We present two strategies to modify the word embedding technique for user embedding. The first is to sequentialize users' ids to get the order of social context. The second is to embed users into a large-sized sliding window of contexts. We conduct extensive experiments on two real data sets from Sina Weibo. Results show that our method is significantly better than the state-of-the-art graph embedding baselines. Its accuracy also outperforms that of the content based approaches.

Keywords: gender prediction; users in social media; social contexts; social representations

Citation: Qian, T., Zhu, P., Li, X. & Sun, D. (2017). Identifying Users' Gender via Social Representations. In iConference 2017 Proceedings, Vol. 2 (pp. 77-86). `https://doi.org/10.9776/17019`

Acknowledgements: The work described in this paper has been supported in part by the NSFC projects (No.61272275, No.61572376, No.61272110, No. 91646206), and the 111 project(B07037).

Contact: {qty,lixuhui}@whu.edu.cn

1 Introduction

Gender prediction has attracted a great deal of research attentions (Cheng, Chen, Chandramouli, & Subbalakshmi, 2009; Mukherjee & Liu, 2010; Otterbacher, 2010; Peersman, Daelemans, & Vaerenbergh, 2011; Filippova, 2012; Bergsma & Durme, 2013; Xiao, Zhou, & Wu, 2013) in recent years due to its potential applications like targeted advertising and personalization. Almost all existing methods use the content texts to build the feature vector for classification (Cheng et al., 2009; Peersman et al., 2011; Filippova, 2012; Bergsma & Durme, 2013). It is often difficult to access the text such as microblogs or reviews due to the restriction of the websites. More importantly, there are a large number of users in social media who register only for browsing, i.e., they do not have contents. For instance, a sample of 1 million users from Sina Weibo in China shows that about 7.4% users do not post any message. For this kind of users, it is impossible to get any content features for analysis.

In this paper, we present a novel approach for gender classification which uses *no* content features. The key idea is to learn the social representation from the relations among users. In social media, there are two fundamental social relations, i.e., following and being followed relations. For simplicity, we will use friendOf to represent both relations, and use friends to represent all users in one user's connection list, including family members, shoolmates, etc. For example, a sequence of "`2: 8 6 4 10`" represents that user 2 has friendOf relations with four users 8, 6, 4, and 10.

Existing methods for representing social relations include the traditional graph based representation (TGR for short) (Culotta, Kumar, & Cutler, 2015) and the recently developed graph embeddings (GE for short) (Perozzi, Al-Rfou, & Skiena, 2014; J. Tang et al., 2015). Basically, each friendOf relation is represented as an edge in the graph. Hence for the above example, we will have four edges, `2-8`, `2-6`, `2-4`, and `2-10`, as illustrated in Figure 1 (a). We argue that some important information are missing from TGR and GE. Note that, besides the *explicit social relations* as direct friendOf connections, there are *implicit social relations* among the friends themselves. For example, user 2 follows users 6 and 10 because they are his/her classmates.

This infers that users 6 and 10 are also classmates. If users 6 and 10 do not follow each other, then there will be no edge between them in the graph.

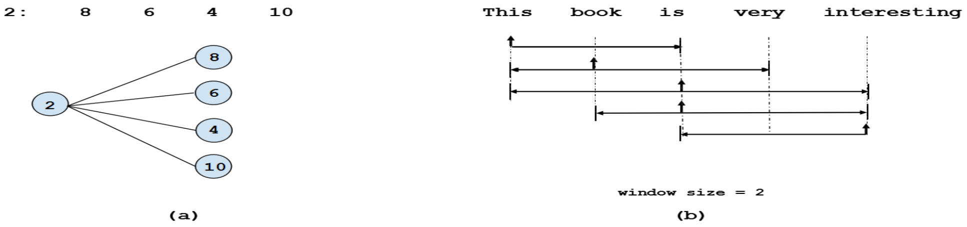

Figure 1: A sample of (a) graph representation, (b) word embedding

Our proposed approach can capture both the implicit and explicit relations. It builds concepts on word embedding (Bengio, Ducharme, Vincent, & Jauvin, 2003; Mnih & Hinton, 2007; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). Figure 1 (b) shows an example for the process of word embedding. Supposing the window size is 2, when modeling a sequence of words (`this`, `book`, `is`, `very`, `interesting`), each word is treated as the current word, and words within the window (left and right) as contexts, and we have five contexts (`book, is`), (`this, is, very`), (`this, book, very, interesting`), (`book, is, interesting`), and (`is, very`). Clearly, compared to the graph representation which only captures the explicit relations, word embedding encodes richer information as it reflects all the implicit relations among words co-occurring in one sentence. Similarly, the implicit social relations among classmates `6` and `10`, corresponding to the word `is` and `interesting`, respectively, can also be captured by this method.

The above example illustrates the improvement of word embedding over the graph representations. However, due to the wide gap between linguistic and social contexts, word embedding has limitations when it is used to encode social relations. In language, syntax governs the sequence of words in a sentence. In contrast, the ordering is missing from the users' ids in social contexts. For example, two classmates 6 and 10 can appear at any position of social contexts. Furthermore, there are a lot of phrases or idioms, shown as local structure in sentences. Hence a small window size like 5 or 10 is usually good enough to capture the local structure in word embedding. However, the related users may be far away from each other in social contexts. To deal with these two problems, we propose two modifications. One is the node sequentializing and the other the large-sized sliding window. The node sequentializing is to map the users' id into a fixed order so as to eliminate the randomness in the neighborhood. The large-sized sliding window aims to enclose users in a long distance yet from same community into one context.

To the best of our knowledge, we are the first to adapt the word embedding approach to exploiting social relations for social representations. Our proposed method has the following key properties.

- It presents a new model to encode social relations which involves only users ids, while most of existing approaches rely on the contents to build feature vectors.
- It captures all kinds of social relations among users, while graph embedding techniques consider only explicit relations between two users.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces our approach for learning social representation for gender prediction. Section 4 introduces the data sets used for evaluation. Section 5 provides experimental results. Section 6 concludes the paper.

2 Related Work

We review the literature in this section, organized by the feature set, the word embedding and graph embedding techniques, and the classification method.

2.1 Feature set

In the area of gender classification, the word or character n-grams are the most widely used features (Peersman et al., 2011; Filippova, 2012; Burger, Henderson, Kim, & Zarrella, 2011; Bergsma & Durme, 2013; Cheng et al., 2009). There are also a number of stylistic features extracted from the content, including the ratio of punctuation, capital letters, unique words (Filippova, 2012), slang words (Goswami, Sarkar, & Rustagi, 2009), word or sentence length (Filippova, 2012; Goswami et al., 2009), conceptual class (Bergsma & Durme, 2013), and the part-of-speech (POS) sequence (Mukherjee & Liu, 2010).

Almost all existing studies rely on the content information. The only exception is the neighbor vector representations (NVR) approach in (Culotta et al., 2015) which directly utilizes the network information. We will use that as a baseline to show the improvements of our embedding method.

2.2 Word embedding and graph embedding

Word embedding has shed lights on many nature language processing (NLP) tasks with the development in deep neural network. The typical techniques include NNLM (Bengio et al., 2003), LBL (Mnih & Hinton, 2007), CBOW and SkipGram (Mikolov et al., 2013). We adapt SkipGram to our task because it performs better than CBOW and also because it significantly speeds up the training process of NNLM and LBL.

Graph embedding is a classic problem. Traditional approaches like Isomap and Laplacian EigenMap have a quadratic time complexity to the number of the nodes. In recent years, researchers proposed GF using stochastic gradient descent (Ahmed, Shervashidze, Narayanamurthy, Josifovski, & Smola, 2013), LINE using edge sampling (J. Tang et al., 2015), and DeepWalk using random walk (Perozzi et al., 2014) for large scale network embedding. Among these, the LINE method achieves the best performance, and hence we use it as one of the baselines.

2.3 Classification method

A number of machine learning approaches have been explored to solve the problem of gender prediction, for instance, SVM (Rao, Yarowsky, Shreevats, & Gupta, 2010; Cheng et al., 2009; Mukherjee & Liu, 2010; Peersman et al., 2011), decision trees (Pennacchiotti & Popescu, 2011; Cheng et al., 2009; Alowibdi, Buy, & Yu, 2013), Naïve Bayes (Mukherjee & Liu, 2010; Goswami et al., 2009; C. Tang, Ross, Saxena, & Chen, 2011; Alowibdi et al., 2013), logistic regression (Bergsma & Durme, 2013; Bamman, Eisenstein, & Schnoebelen, 2014), the Winnow algorithm (Burger et al., 2011; Schler, Koppel, Argamon, & Pennebaker, 2005), and the maximum entropy learner (Filippova, 2012).

The classification method is not the focus of this paper. In our study, we choose to use LR as our base classifier since LR is not as sensitive to parameters as SVM and also because it performs well.

3 Learning Social Representation

In social media, users are connected with their family members, friends, schoolmates, colleagues, or people with similar interests. All these connections (neighbors) form *social contexts*. Furthermore, being friendOf with a same user, these connections may belong to a same community. For example, a user a follows her classmates b and c. Then "b and c" are the user a's social context, and users a, b, and c form a community "classmate". We can further deduce that if there exists another user d in this class, then d may have a social context of "a, b, and c". The more times users appear in the same social contexts, the stronger relations are there among these users, and the larger probability they belong to the same community. Such an observation inspires us to borrow ideas from SkipGram (Mikolov et al., 2013), a recently developed word embedding technique which captures the semantic and syntactic relations among words.

3.1 Preliminary on SkipGram

The objective of SkipGram is to maximize the co-occurrence probability among the words that appear within a window in a sentence. More formally, we can define the objective function as:

$$L = \sum_{w \in C} logp(Context(w)|w)) = \sum_{w \in C} log \prod_{u \in Context(w)} p(u|w) \tag{1}$$

where C is the corpus used for training. By applying the Huffman tree based hierarchical softmax (Mikolov et al., 2013), we can rewrite $p(u|w)$ as:

$$p(u|w) = \prod_{j=2}^{l^u} p(d_j^u|v(w), \theta_{j-1}^u) \tag{2}$$

$$p(d_j^u|v(w), \theta_{j-1}^u) = [\sigma(v(w)^T \theta_{j-1}^u)]^{1-d_j^u} \cdot [1 - \sigma(v(w)^T \theta_{j-1}^u)]^{d_j^u} \tag{3}$$

where $v(w)$ is the d-dimension vector for the central word w, and d_j^u and θ_j^u is the Huffman code (either 0 or 1) and the d-dimension vector for the j_{th} node on the path p^w, respectively. Equation (1) can then be optimized using gradient descending technique. The procedure is shown in Algorithm 1.

Algorithm 1: SkipGram

```
SkipGram($\Phi, b, Context(u_i), s, \eta$)
  for each $v_j \in Context(u_i)$
      for each $u \in Context(u_i)[b, b+s]$
          $L(\Phi) = log \prod_{u \in Context(u_i)} p(u|\Phi(v_j))$
          $\Phi = \Phi - \eta \frac{\partial L}{\partial \Phi}$
      end for
  end for
```

3.2 Adapting SkipGram to user embedding

SkipGram is designed for word embedding. A typical scenario to use SkipGram is upon a corpus, where each word is naturally embedded in a paragraph or document. However, as analyzed in the previous section, the users in social media do not occupy such an characteristic. Furthermore, the users with local structure or community may be far apart from each other. Hence we present two strategies to adapting SkipGram to user embedding, i.e., node sequentializing and the large-sized sliding window.

3.2.1 Node sequentializing

Node sequentializing is the process of identifying, for each social context, a sequence of nodes for which the neighbors of a node are created, like the syntax governing the sequence of words in a sentence. The simplest sequentializing is just to use *the nature order* of ids. For example, user 1 precedes user 2, and user 2 precedes user 3. This method can eliminate the randomness of neighbors. However, the id information is irrelevant to the inherent structure of the network. Hence we present the following degree-based sequentializing method.

Given any two nodes i and j, and their k-th layer of neighbors $\aleph_k(i)$ and $\aleph_k(j)$ in Graph G, the degree-based sequentializing defines a total order $\succ$ on nodes in G which uniquely determines the position O of a node i in a sequence, such that:

a. $O(i) \succ O(j)$ iff. $d(i) > d(j)$;

b. $O(i) \succ O(j)$ iff. $d(i) = d(j)$, and $d(\aleph_1(i)) > d(\aleph_1(j))$;

c. $O(i) \succ O(j)$ iff. $d(i) = d(j)$, $d(\aleph_k(i)) = d(\aleph_k(j))$ ($k = 1..m-1$), and $d(\aleph_m(i)) > d(\aleph_m(j))$.

The $d(i)$ function denotes mapping the node id to the degree of this node. The basic idea is to sort the ids by their degrees. If the degrees of two nodes are equal, then incrementally compare their k-th neighbors' degrees until a total order is given.

3.2.2 large-sized sliding window

The *large-sized sliding window* strategy is presented to deal with the problem caused by the randomness of user id, which is automatically assigned by the system and normally correlated with the time users start to use the social media. This means that users' connections may not have the adjacent neighbor ids. When SkipGram is used in the NLP applications, the window size is often set to 5 or 10. This is definitely not suitable for user embedding because the number of friends is usually quite large. Hence we set the window size s in SkipGram to a large value.

3.3 Algorithm for learning social representation

We can now learn the social representation by applying the SkipGram to the above prepared social contexts. The entire procedure, called as User Embedding (UE), is shown in Algorithm 2.

Algorithm 2: User Embedding (UE)

Input: The set of users $U = \{u_i\}$ and the friends of users $\{u_i, F(u_i) = \{f_{i1}, ...f_{in}\}\}$
Parameters:, the window size s, the dimensionality of user embedding d, and the learning ratio η
Output: matrix of user representations $\Phi \in \Re^{|U| \times d}$
Steps:

```
1. for each user $u_i$
2.     $A(u_i) = \{u_i\} \bigcup F(u_i)$
3. end for
4. Sequentializing ids in $F(u_i)$ and $A(u_i)$
5. for each user $u_i$
6.     for the $jth$ friend (j = $i_1..i_n$) in $F(u_i)$
7.         SkipGram($\Phi, j, F(u_i), s, \eta$)
8.     end for
9.     SkipGram($\Phi, 1, A(u_i), |A(u_i)|, \eta$)
10. end for
```

In Algorithm 2, lines 1-3 initiate the explicit social contexts. Line 4 sequentializes the ids. Lines 5-10 build the user embedding. Specifically, lines 6-8 iterate on the implicit social contexts and line 9 on the explicit social contexts.

There are three parameters tunable in the UE algorithm: η the learning ratio, d the dimensionality of embedding vector, and s the size of contexts (also called the window size). Among which, η is related to the training speed and we just use the default setting. We investigate the effects of d and s in the experimental part.

4 Data sets

4.1 Data collection

The data is collected from Sina Weibo, which is one of the largest micro-blogging services in China. Each user in Sina Weibo has a profile, which has several fields, such as userid, screen name, gender, tags, description, the number of followers, followees, and messages. Table 1 shows a sample profile of a celebrity in Sina Weibo.

Most of the fields in the user's profile are optional. Many users choose to leave them blank, and many users do not post any microblogs either. However, due to the very nature of social media, the users

userid	1749127163
screenname	Leijun
gender	male
the number of messages	4353
the number of followers	11,979,274
the number of followees	921
tags	angel investment, Mi mobile phone, we love mi-chat
description	CEO of Xiaomi, CEO of Kingsoft

Table 1: A sample user profile in Sina Weibo

intend to connect with others. This results in a number of connections for each user. We can crawl the uids of friends through the API provided Sina, which can be used for our experiments.

We start from a public domain data set [1] including the profile information of 1 million users. From which, we construct two data sets. One is the mute celebrities and the other the ordinary users.

4.2 Mute celebrities

The data set of mute celebrities (MC for short) contains 1280 users (640 female and 640 male users, respectively) who have at most 5 microblogs. We build such a data set due to the reasons below.

- At the beginning, we intend to choose the real mute celebrities from the original public data. However, we only find 21 celebrities meeting this requirement. This is too sparse for performing experiments. Hence we have to expand the data set by relaxing the minimum number of posts to 5.
- We select the mute celebrities rather than the mute ordinary users since it is extremely hard to examine their gender if there is no additional information (e.g., microblogs, photos) for lurkers. In contrast, the gender of celebrities has been verified by Sina and hence there is no need to manually check gender.

This data set is used to validate the effectiveness of our method on mute users, which is the initial objective of our study.

4.3 Ordinary users

The data set of ordinary users (OU for short) contains 400 users (200 female and 200 male users, respectively) who have at least 10 microblogs. The number of ordinary users is much less than that of mute celebrities because of the labor-intensive procedure of manually checking. In order to verify the users' gender in this data set, we recruit three undergraduate students to manually check the data in order to ensure 1) the account is not a spammer or implicit enterprise users like the owner of micro-shops, 2) the user's gender is real, which is done by keeping only those with more than agreed labels from two students.

This data set is used to evaluate how our method performs on ordinary users. Moreover, we wish to compare our user embedding method with the content based approaches using features extracted from microblogs. This is why we set the condition of the minimum number of 10 microblogs for this data set.

For both the MC and OU data sets, we download up to 500 uids of the friends through the friends API from Sina. By using these uids, we build the implicit and explicit social contexts and construct the embedded uid vectors for our approach. This also forms the first layer graph. We further collect two-hop friends (neighbors' neighbors) using the same script to construct the second layer graph. This additional part is necessary for the graph-based baselines (see below). Table 2 shows the statistic for two data sets, where the edge and degree in the first and second layer of graph is represented as the superscript of 1 and 2, respectively.

We find from Table 2 that the average degree of MC is much smaller than that of OU in the first layer graph. This can be due to the difference between mute and ordinary users. The lurkers are not only silent in most of times but also inactive in social networking. Also note that the average degree in the

[1] http://www.nlpir.org/?action-viewnews-itemid-232

Data set	UserNum^1	EdgeNum^1	AvgDegree ^{1}	UserNum^2	EdgeNum^2	AvgDegree^2
OU	400	66686	166.72	51152	1184586	23.16
MC	1280	125340	97.92	87202	2722699	31.22

Table 2: The statistic for two data sets

second layer graph is significantly smaller than that in the first layer. The reason is that due to the resource restriction, we only keep the edges in second layer which already have one node in the first layer. The users (nodes) of the remaining edges are two hops away from the original 400 and 1280 users and thus are out of our considerations.

5 Experiments

We conduct experiments on two real data sets as introduced in the previous section. The users are randomly divided into five parts. We perform 5-fold cross validation and the results are averaged over five folds. We use the accuracy as the evaluation metric since both data sets are balanced on two classes.

5.1 Effects of window size

The window size and dimensionality are two key parameters in user embedding. We first investigate the effects of window size and show the results in Figure 2.

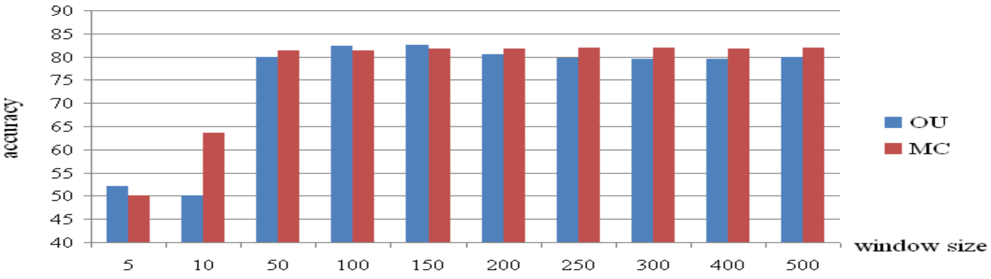

Figure 2: Effects of window size

It is clear that the best performance is achieved at the window size of 150 and 250 for the OU and MC data set, respectively, which is much larger than the small window size of 5 or 10 for the language contexts. Indeed, the accuracy on window size 5 on two data sets is 52.00 and 50.25. Both are the worst. This shows that the large-sized window strategy is appropriate for the user embedding problem. We will use the best window size as our default setting in the following experiments.

5.2 Effects of dimensionality

We investigate the effects of dimensionality by scaling it from 50 to 300. The results are shown in Table 3.

	50	100	150	200	250	300
OU	79.75	82.50	82.00	82.50	**83.00**	82.50
MC	81.88	**82.03**	81.48	81.95	81.25	81.88

Table 3: Effects of dimensionality

From Table 3, we find the fluctuation of accuracy on MC is less obvious than that on OU. The largest change is 0.78%, showing that it is not very sensitive to the dimensionality. In addition, it can be seen that the best performance is 83.00% and 82.03% on OU and MC, respectively, reached at the dimensionality of

250 and 100. However, for the fair comparison with the word embedding and graph embedding approaches, we set the dimensionality to 100 in the following experiments, which is the same as those used in (Mikolov et al., 2013; J. Tang et al., 2015).

5.3 Effects of sequentializing

We evaluate the effects of three types of node sequentializing, i.e., by the random order, by the nature order, and by the degree order. The results are shown in Table 4. Note that all the results are under the best window size for each method. It can be seen that the accuracy on MC by the random order is better than that on OU. This contradicts with that by the nature and degree order, showing that the random order is not stable. We can also see that the results by the degree order are the best. The reason may be that in social network, well-connected nodes tend to connect to each other, known as the rich-club phenomenon (Colizza, Flammini, Serrano, & Vespignani, 2006). Using the degree order helps finding the inherent structure in social contexts and thus improves the performance.

	random	nature	degree
OU	78.00	81.00	**82.50**
MC	79.61	79.53	**82.03**

Table 4: Effects of sequentializing

5.4 Comparison with baselines

We conduct extensive comparison experiments on six baselines, including content-based, graph-based, graph embedding, and user embedding approaches. The description about the baselines are as follows.

- Word Frequency (WF): This baseline uses the contents from microblogs. Each word in the microblog is represented as a vector with the dimension as x:y, where x is word id and y its frequency. This is the most widely used representation in gender classification.
- Word Embedding (WE): This baseline uses the words in users and their friends as the corpus to get the distributed representations of words. Following the practice in (Mikolov et al., 2013), we set the dimensionality to 200, and use the hierarchical softmax for approximation.
- Original Graph Representations (TGR): This is the traditional one-hot representation with the entry in users' vector as x:y, where x is the user id and y is 1 or 0, standing for whether x appears in this user's neighbor.
- Neighbor Vector Representations (NVR): This baseline represents each user as a neighbor vector with the dimension as x:y, where x is the two-hop neighbor node and y stands for the fraction of its followers that are friends with each of its neighbors. For classification, we strictly follow the settings in (Culotta et al., 2015).
- Graph embedding method (LINE): This contains two baselines LINE_1 and LINE_{1+2}, which uses the users, the one- and two-hop friends to construct the network for graph embedding, respectively. We use the default settings in (J. Tang et al., 2015), i.e., the dimension size is 100, the number of negative samples is K=5, and the total number of samples is T=10 billion.

The results are shown in Table 5. We have the following important notes.

a. Our proposed UE method is the best among all approaches. Its improvements over other baselines are all significant under the 0.05 significant test. Firstly, it achieves a huge enhancement over the graph based NVR baseline with a 16.50% and 23.59% improvement on OU and MC. Secondly, it outperforms two graph embedding approaches by a large margin. For example, the accuracy on OU grows from 73.00% (LINE_1) to 82.50%. The performances of LINE_{1+2} on both OU and MC are much worse than those of our UE as well. This clearly demonstrates that UE is more effective and resource efficient than the state-of-the-art

	content based		graph based		graph embedding		user embedding
	WF	WE	TGR	NVR	LINE_1	LINE_{1+2}	UE
OU	76.00	74.75	79.53	66.00	73.00	71.50	**82.50**
MC	*	*	76.00	58.44	74.06	68.44	**82.03**

Table 5: Comparison with baselines

graph embedding approach LINE. Thirdly, it is much better the content based approaches WF and WE. This is a very strong indication of the potential application scenario of UE. No matter whether there are text information, UE is a good choice for classification.

b. The traditional one-hot graph representation TGR is the second best. This suggests that a users gender is highly correlated with his/her friends. This finding is interesting, indicating that the majorities in female users' social networks are female. The same conclusion also holds for the male users. Furthermore, TGR outperforms the graph embedding LINE, suggesting that graph embedding may incur information loss. In contrast, our user embedding approach UE models both the implicit and explicit social relations rather than building a graph for explicit connections and thus is a better representation. We notice that the other graph based method NVR is the worst. The reason may be that NVR only keeps the indirect relations between users.

c. The performance of graph embedding method LINE_{1+2} is worse than LINE_1, different from that in (J. Tang et al., 2015). The reason may be arisen from that we do not include the edges whose nodes are two-hops away from the original users. Remember that the average degree in second layer graph is much smaller than that in the first layer, especially on the OC data set. This finding is important in that the second proximity may hurt the performance unless enough nodes and edges are supplemented. However, this needs extra overheads. We also find that WF is better than WE, suggesting that the content based approach does not benefit from the word embedding technique. This can be due to the diverse topic and vocabulary in microblogs.

6 Conclusion

We present a novel social representation to capture both the explicit and implicit relations among users in social media. By modifying the word embedding technique to exploit the social contexts, the proposed approach achieves very accurate results on gender classification. We conduct extensive experiments on two real data sets from Sina Weibo. The results show that our UE method is significantly better than both the traditional grpah based approaches and the state-of-the-art graph embedding algorithms. This clearly demonstrates that our approach is extremely useful when the texts are unavailable. Further more, our method also outperforms the content based approaches. This strongly indicates that our method has general applications to all types of users.

In the future, we plan to conduct more experiments on data sets from Twitter or Facebook. We also plan to investigate how our method works on other user profiling tasks.

References

Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., & Smola, A. J. (2013). Distributed large-scale natural graph factorization. In *Proc. www* (p. 37-48).

Alowibdi, J. S., Buy, U. A., & Yu, P. (2013). Empirical evaluation of profile characteristics for gender classification on twitter. In *Proc. of the 12th icmla* (p. 365-369).

Bamman, D., Eisenstein, J., & Schnoebelen, T. (2014). Gender identity and lexical variation in social media. *Journal of Sociolinguistics*, *18*, 135-160.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *JMLR*, *3*, 1137-1155.

Bergsma, S., & Durme, B. V. (2013). Using conceptual class attributes to characterize social media users. In *Proc.of acl* (p. 710-720).

Burger, J. D., Henderson, J., Kim, G., & Zarrella, G. (2011). Discriminating gender on twitter. In *Proc. of emnlp* (p. 1301-1309).

Cheng, N., Chen, X., Chandramouli, R., & Subbalakshmi, K. P. (2009). Gender identification from e-mails. In *Proc. of cidm* (p. 154-158).

Colizza, V., Flammini, A., Serrano, M., & Vespignani, A. (2006). Detecting rich-club ordering in complex networks. *Nature Physics*, *2*, 110-115.

Culotta, A., Kumar, N., & Cutler, J. (2015). Predicting the demographics of twitter users from website traffic data. In *Proc. of aaai* (p. 72-78).

Filippova, K. (2012). User demographics and language in an implicit social network. In *Proc. of emnlp-conll* (p. 1478-1488).

Goswami, S., Sarkar, S., & Rustagi, M. (2009). Stylometric analysis of bloggersÂąÂŕ age and gender. In *Proc. of icwsm* (p. 214-217).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, *abs/1310.4546*.

Mnih, A., & Hinton, G. (2007). Three new graphical models for statistical language modelling. In *Proc. of icml* (p. 641-648).

Mukherjee, A., & Liu, B. (2010). Improving gender classification of blog authors. In *Proc. of emnlp* (p. 207-217).

Otterbacher, J. (2010). Inferring gender of movie reviewers: Exploiting writing style, content and metadata. In *Proc. of cikm* (p. 369-378).

Peersman, C., Daelemans, W., & Vaerenbergh, L. V. (2011). Predicting age and gender in online social networks. In *Proc. of smuc* (p. 37-44).

Pennacchiotti, M., & Popescu, A.-M. (2011). A machine learning approach to twitter user classification. In *Proc. of fifth international aaai conference on weblogs and social media* (p. 281-288).

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proc. of sigkdd* (p. 701-710).

Rao, D., Yarowsky, D., Shreevats, A., & Gupta, M. (2010). Classifying latent user attributes in twitter. In *Proc. of smuc* (p. 37-44).

Schler, J., Koppel, M., Argamon, S., & Pennebaker, J. W. (2005). Effects of age and gender on blogging. In *Proc. of aaai spring symposium on computational approaches for analyzing weblogs* (p. 199-205).

Tang, C., Ross, K., Saxena, N., & Chen, R. (2011). What's in a name: A study of names, gender inference, and gender behavior in facebook. In *Proc. of snsmw.*

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). Line: Large-scale information network embedding. In *Proc. of www* (p. 1067-1077).

Xiao, C., Zhou, F., & Wu, Y. (2013). Predicting audience gender in online content-sharing social networks. *JASIST*, *64*, 1284-1297.