Biased Random Walk based Social Regularization for Word Embeddings

Ziqian Zeng[1*], **Xin Liu**[1,2*], **and Yangqiu Song**[1]
[1]Department of CSE, The Hong Kong University of Science and Technology.
[2]School of Data and Computer Science, Sun Yat-sen University.
zzengae@cse.ust.hk, xliucr@connect.ust.hk, yqsong@cse.ust.hk

Abstract

Nowadays, people publish a lot of natural language texts on social media. Socialized word embeddings (SWE) has been proposed to deal with two phenomena of language use: everyone has his/her own personal characteristics of language use and socially connected users are likely to use language in similar ways. We observe that the spread of language use is transitive. Namely, one user can affect his/her friends and the friends can also affect their friends. However, SWE modeled transitivity implicitly. The social regularization in SWE only applies to one-hop neighbors and thus users outside the one-hop social circle will not be affected directly. In this work, we adopt random walk methods to generate paths on the social graph to model the transitivity explicitly. Each user on a path will be affected by his/her adjacent user(s) on the path. Moreover, according to the update mechanism of SWE, fewer friends a user has, fewer update opportunities he/she can get. Hence, we propose a biased random walk method to provide these users with more update opportunities. Experiments show that our random walk based social regularizations perform better on the sentiment classification task.

1 Introduction

Word embeddings [Mikolov *et al.*, 2013a; Mikolov *et al.*, 2013b; Pennington *et al.*, 2014] have been widely used in natural language processing tasks. When analyzing the natural language use on social media platforms or consumer review websites such as Facebook, Twitter, and Yelp, we observe that everyone has his/her own personal characteristics of language use. For example, everyone has his/her own preference for diction and expression method. The hypothesis of distributional representation states that words with similar meanings tend to appear in similar contexts [Harris, 1954]. For different people, contexts around a word would be different due to their personal characteristics of language use. Hence, there is a need to consider personal characteristics of language use in word embeddings. We also observe that socially connected people tend to use language in similar manners. As indicated by [Hovy, 2015], language use can be affected by demographic factors such as age, gender, race, geography and so on [Rosenthal and McKeown, 2011; Eckert and McConnell-Ginet, 2003; Green, 2002; Trudgill, 1974; Fischer, 1958; Labov, 1963]. For example, scientists on social media may mention more scientific and hi-tech related terms while movie stars may mention more about entertainment news. Moreover, some groups of people say "Y'gotta do it the right way." while others say "You have to do it the right way." It is not straightforward to access demographic information, but it is reasonable to believe that friends on social media tend to have some demographic factors in common. Hence, it is reasonable to incorporate this phenomenon when we consider personalized language use.

Socialized word embeddings (SWE) [Zeng *et al.*, 2017] has been proposed to deal with aforementioned two phenomena with two modifications of word embeddings: personalization and socialization. For personalization, SWE adopted word2vec [Mikolov *et al.*, 2013a] as the base model and applied personalization to words by introducing a user vector for each social user. The word representation for each social user is the summation of a global word vector and a local user vector. For socialization, SWE added a social regularization term to impose user vectors between friends to be similar. It was shown that SWE can improve word embeddings on word representation learning for social media sentiment analysis [Zeng *et al.*, 2017]. However, there are still two problems.

First, we observe that the spread of language use is transitive. Namely, one user can affect his/her friends and the friends can further affect their friends. However, the social regularization in SWE only applies to pairwise friends, i.e., one-hop neighbors. The users outside the one-hop social circle will not be updated until one of their one-hop neighbors is considered, so the transitivity is modeled implicitly. It would be more effective if we can model the transitivity more explicitly in the social regularization. Second, since there are in general more texts input by users with more friends than texts input by users with fewer friends, when optimizing the social regularization, users with fewer friends will be updated much less than users with more friends. It would be helpful if we can find a better way to model users with fewer friends to train their user embeddings more frequently.

*Equal contribution. This work was done when Xin Liu was a research assistant at HKUST.

To solve the above two problems, in this paper, we propose to use random walk based methods for the social regularization, which can generate paths to explicitly model the transitivity and can control the process of user sampling. The nodes in the path are analogous to affected users during the propagation of language use. Each user on a path will be affected by his/her adjacent user(s) on the path. The change triggered by start user will pass along the path and all users will be updated accordingly. We propose to use both *first-order* and *second-order random walk* based regularizations. In first-order random walk, a random walker moves to next node based on the last node while in second-order random walk [Grover and Leskovec, 2016], it relies on both the last and the second last nodes. Experiments show that social regularizations using aforementioned random walk methods perform better than SWE. Moreover, we propose a *biased random walk* based regularization which introduces a bias coefficient to adjust transition probabilities. The bias coefficient is associated with the number of friends of a user so that users with fewer friends can be sampled more frequently. Experiments show that our random walk based social regularizations perform better on sentiment classification task on Yelp review datasets. The code is available at https://github.com/HKUST-KnowComp/SRBRW.

2 Related Work

In this section, we review our related work in two categories.

2.1 Personalization and Socialization in Language Modeling

Language models are fundamental to natural language processing. Users of search engines often have different search purposes even when they submit the same query. To consider personalization, personalized language models have been developed by [Croft *et al.*, 2001; Song *et al.*, 2010; Sontag *et al.*, 2012] and applied to personalized web search. However, some users may not have sufficient corpora to train personal language models. Hence, socialized language models [Vosecky *et al.*, 2014; Huang *et al.*, 2014; Yan *et al.*, 2016] were proposed to deal with the sparsity problem. Word embeddings are also important in NLP and can be easily integrated into downstream tasks. Socialized word embeddings [Zeng *et al.*, 2017] was proposed to deal with phenomena of language use. Everyone has his/her own personal characteristics of language use and socially connected users are likely to use language in similar ways.

2.2 Random Walk Methods

A random walk is a stochastic process which consists of movements from a node to another adjacent node. Each movement relies on previous node(s) and associated transition probabilities. Random walks have been applied in different research fields [Weiss, 1983]. Network representation learning is one of successful applications. Inspired by word2vec [Mikolov *et al.*, 2013a; Mikolov *et al.*, 2013b], DeepWalk [Perozzi *et al.*, 2014] adopted a first-order random walk method to generate walks by treating walks as the equivalent of sentences. The transition probabilities used in DeepWalk are uniform. To capture a diversity of network structures, node2Vec [Grover and Leskovec, 2016] used second-order random walks to learn better representation. Node2vec used two tunable parameters to flexibly adjust exploration strategies.

3 Methodology

We first briefly introduce the SWE model [Zeng *et al.*, 2017] and then introduce our random walk methods.

3.1 Socialized Word Embedding (SWE)

Suppose there are N users $u_1, \ldots, u_N$ in a social network. A user u_i's one-hop neighbors set is denoted as $\mathcal{N}_i = \{u_{i,1}, \ldots, u_{i,N_i}\}$, where N_i is the number of one-hop neighbors of user u_i. We aggregate all documents published by user u_i as a corpus $\mathcal{W}_i$. In CBOW [Mikolov *et al.*, 2013a] based SWE model, given a sequence of training words, the first objective is to minimize the negative log-likelihood:

$$\mathcal{J}_1 = -\sum_{i}^{N} \sum_{w_j \in \mathcal{W}_i} \log P(w_j | \mathcal{C}(w_j, u_i)), \tag{1}$$

where w_j is the predicted word and $\mathcal{C}(w_j, u_i)$ is a collection of context words around w_j. To apply personalization to a word w_j, SWE represented a word as $\mathbf{w}_j^{(i)} = \mathbf{w}_j + \mathbf{u}_i$, where $\mathbf{w}_j \in \mathbb{R}^d$ is the global word embedding and $\mathbf{u}_i \in \mathbb{R}^d$ is the local user embedding for user u_i. The representation of context words $\mathcal{C}(w_j, u_i)$ is $\{\mathbf{w}_{j-c}^{(i)}, \ldots, \mathbf{w}_{j+c}^{(i)}\}$, where c is the half window size.

A socialized regularization term is added to consider the second phenomenon:

$$\mathcal{J}_2 = \sum_{i}^{N} \sum_{u_j \in \mathcal{N}_i} \frac{1}{2} ||\mathbf{u}_i - \mathbf{u}_j||_2^2, \tag{2}$$

where u_j is a friend of user u_i. This regularization aims to force user vectors between friends to be similar.

By combining two parts, the final objective of SWE is

$$\begin{aligned} &\mathcal{J} = \mathcal{J}_1 + \lambda \mathcal{J}_2 \\ &s.t.\ \forall\ u_i,\ r_1 \leq ||\mathbf{u}_i||_2 \leq r_2, \end{aligned} \tag{3}$$

where λ is a trade-off parameter, r_1 and r_2 are a lower bound and upper bound for $\mathbf{u}_i$'s L_2-norm respectively. r_1 can avoid the situation where user embeddings might collapse to zero vector and thus SWE will degenerate to word2vec [Mikolov *et al.*, 2013a] while r_2 can prevent user embeddings dominating global word embeddings.

In SWE, when a document published by u_i is observed, $\mathbf{u}_i$ will be updated due to $\mathcal{J}_1$, and user embeddings of u_i's friends $\mathcal{N}_i$ will also be updated due to $\mathcal{J}_2$. But users outside this one-hop social circle will not be updated until one of texts published by themselves or their one-hop neighbors is observed.

Algorithm 1 SWE with regularization using Random Walks.

Input: User set $\mathcal{U} = \{u_1, u_2, ..., u_N\}$, where each user has a corpus $\mathcal{W}_i = \{d_{i,1}, \ldots, d_{i,M_i}\}$ and M_i is the number of documents written by u_i, maximum iteration T, learning rate on Eq. (1) η_1, learning rate on Eq. (2) η_2, trade-off parameter λ, n_i paths that random walks generate for user u_i, path length l, return parameter p, in-out parameter q, restart rate α, bias weight β.

```
for iter ← 1 to T do
  for all u_i ∈ U do
    for all d_{i,j} ∈ W_i do
      for all w_k ∈ d_{i,j} do
        w_k := w_k − η_1 · ∂J_1/∂w_k
        u_i := u_i − η_1 · ∂J_1/∂u_i
      end for
      walks = RandomWalk(u_i, n_i, l, p, q, α, β) following Eqs. (5)-(7)
      for walk ∈ walks do
        SocialRegularization(walk, η_2, λ)
      end for
    end for
  end for
end for
```

Algorithm 2 SocialRegularization

Input: $walk = \{u_0, ...u_{l-1}\}$, learning rate η_2, parameter λ.

```
for all u_i ∈ walk do
  if i = 0 then
    N = {u_{i+1}}
  else if i = l − 1 then
    N = {u_{i−1}}
  else
    N = {u_{i−1}, u_{i+1}}
  end if
  for all u_j ∈ N do
    u_i := u_i − η_2 λ · (u_i − u_j)
    u_j := u_j − η_2 λ · (u_j − u_i)
  end for
end for
```

3.2 Random Walk based Social Regularization

As we explained in the introduction, there are two major problems with the above social regularization framework: implicit modeling on transitivity and lack of concern of users with fewer friends. To remedy the problems, we propose to augment the social regularization with a random walk based approach. Intuitively, instead of imposing a regularizer within the one-hop social circle, we sample a set of random walks starting from the user. Then we impose the regularizer over all the sampled users in a path to explicitly model the transitivity. To emphasize the users with fewer friends, we also propose a biased random walk to sample more these users in the path. So we still follow the SWE framework but use random walks based regularization.

Here, we only consider random walk methods on the unweighted and undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Given a source node s, random walk methods aim to generate a walk of fixed length l. Let c_i denote the i th node in the walk, starting with $c_0 = s$. Suppose a random walker has just traversed node c_{i-2}, and currently resides in node c_{i-1} and will probably move to c_i. The transition probability $P(c_i|c_{i-1}, c_{i-2})$, can be computed as follows,

$$P(c_i|c_{i-1}, c_{i-2}) = \begin{cases} \frac{\pi_{c_i|c_{i-1},c_{i-2}} \cdot w_{c_{i-1}c_i}}{Z}, & \text{if } (c_{i-1}, c_i) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where Z is a normalizing factor, and $\pi_{c_i|c_{i-1},c_{i-2}}$ is the unnormalized transition probability to c_i given c_{i-1} and c_{i-2}, and $w_{c_{i-1}c_i}$ is the weight of edge (c_{i-1}, c_i). In an unweighted graph, $w_{c_{i-1}c_i} = 1$. We assume our random walk is a Markov chain with stationary transition probabilities. Based on the transition probability $P(c_i|c_{i-1}, c_{i-2})$, we can sample a sequence of nodes, starting with $c_0 = s$.

After generating node sequences using random walks, we apply the social regularization as Eq. (2) to users. The detailed algorithm is shown in Algorithms 1 and 2. Different from SWE, for each node in generated walks, a node in the path will be updated by its adjacent nodes. In this way, we can model the transitivity more explicitly than SWE.

Now the remaining problem is how to formulate the unnormalized transition probability $\pi_{c_i|c_{i-1},c_{i-2}}$. We will describe it in the following to show differences in three random walk methods.

First-order Random Walk

In the first-order random walk (FRW), a random walker moves to next node based on the last node. The unnormalized transition probability $\pi_{c_i|c_{i-1},c_{i-2}} = \pi_{c_i|c_{i-1}}$ is computed as follows,

$$\pi_{c_i|c_{i-1}} = \begin{cases} 1, & \text{if } (c_{i-1}, c_i) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

This method will encourage a very deep walk, which means the random walker tends to go far away from the source node. From the perspective of propagation of language use, the start user might have little influence on the users who are far away from him/her. On the contrary, the start user will have more impacts on his/her close neighbors. This random walk method is not efficient as it tends to sample a lot of remote users.

Second-order Random Walk

In the second-order random walk (SRW), a random walker moves to next node based on the last node and the second last node. The unnormalized transition probability introduced in [Grover and Leskovec, 2016] is computed as follows,

$$\pi_{c_i|c_{i-1},c_{i-2}} = \begin{cases} \frac{1}{p}, & \text{if } d_{c_{i-2}c_i} = 0 \\ 1, & \text{if } d_{c_{i-2}c_i} = 1 \\ \frac{1}{q}, & \text{if } d_{c_{i-2}c_i} = 2 \end{cases}, \tag{6}$$

where $d_{c_{i-2}c_i}$ is the shortest path distance between nodes c_{i-2} and c_i, p is the return parameter controlling the likelihood of immediately revisiting last node c_{i-2} in the walk, and q is the in-out parameter controlling the walker to explore remote friends or close friends. When $p < min(1, q)$, the walker tends to revisit the last node. When $q < min(1, p)$, it tends to explore remote neighbors. When $1 < min(p, q)$, it tends to explore mutual friends.

Compare to the FRW which encourages a very deep walk, p and q allow this method to explore more close neighbors if setting $p < q$. If we let $p = q = 1$, SRW is the same as FRW. We use the alias sampling algorithm introduced in [Grover and Leskovec, 2016] to sample c_i efficiently in $\mathcal{O}(1)$ time given c_{i-2} and c_{i-1}.

Biased Second-order Random Walk

According to update mechanism of SWE, the fewer friends a user has, the fewer update opportunities he/she can get. To tackle this problem, we propose a biased second-order random walk (BRW) to sample more users who have fewer friends by introducing a bias coefficient to adjust transition probabilities. The bias coefficient is associated with the number of friends of a user. We define the unnormalized transition probability $\pi'_{c_i|c_{i-1},c_{i-2}}$ as follows,

$$\pi'_{c_i|c_{i-1},c_{i-2}} = \varphi(c_i) \cdot \pi_{c_i|c_{i-1},c_{i-2}}, \tag{7}$$

where $\varphi(\cdot)$ is the bias coefficient and $\pi_{c_i|c_{i-1},c_{i-2}}$ is the same as Eq. (6). The transition probability is biased by a factor of $\varphi(\cdot)$ which differs from one node to another. We define $\varphi(x)$ as follows,

$$\varphi(x) = \frac{\frac{1+\beta}{d_x}}{\frac{1+\beta}{d_x} + \frac{1}{d}} \cdot \frac{(1+\beta)+1}{1+\beta}, \tag{8}$$

where $d = |\mathcal{E}|/|\mathcal{V}|$ is averaged number of friends of a social network, d_x is the number of friends of node x, and β is a parameter to adjust the shape of $\varphi(x)$ (see Figure 1(a)).

When $d_x < d$, we have $\varphi(x) > 1$, which means when the number of friends of node x is below the average, the walk tends to move to x with a larger probability than before. When $d_x > d$, we have $\varphi(x) < 1$, which means when the number of friends of node x is above the average, the walk tends to move to x with a smaller probability than before. when $d_x = d$, we have $\varphi(x) = 1$, which is exactly the same with the aforementioned probability in the second-order random walk. In Figure 1(a), we can see that smaller β leads to larger adjustment and encourages the random walker to move to users who have fewer friends with larger probability. In particular, when $\beta = 0$, $\varphi(x) = (\frac{2}{\frac{1}{d_x}+\frac{1}{d}})/d_x$ is the ratio of the harmonic mean to the number of friends of node x.

In practice, methods [Perozzi *et al.*, 2014; Grover and Leskovec, 2016] using random walks will generate n paths with fixed length l at a time. To sample more users who have fewer friends, we allow the number of paths to vary with the number of friends. We define the number of paths which associated with a source node as follows,

$$n_s = \varphi(s) \cdot n, \tag{9}$$

where $\varphi(\cdot)$ refers to the same definition in Eq. (8). In this way, a user with fewer friends will generate more paths.

As indicated by [Pan *et al.*, 2004], the restart mechanism is a useful technique in random walks. It is reasonable to introduce the restart mechanism to our biased random walk method since it will encourage more close neighbors. The effect of restart is similar to a small value of p in SRW. The difference is that the restart is more flexible since a walker can move back to source node s at any movement no matter how far away from it. Before a walker makes each movement, restart mechanism allows the walker to determine whether to move back to source node s with a probability as follow,

$$\alpha_s = \varphi(s) \cdot \alpha, \tag{10}$$

where $\varphi(\cdot)$ refers to the same definition in Eq. (8), and α is the initial restart rate. Compared with opinion leaders in social media, users who have fewer friends might have fewer impacts on the users far away from him/her, so it is reasonable to allow them to restart with larger probability.

Dataset	Yelp Round 9	Yelp Round 10
#Users	1,029,432	1,183,361
#Reviews	4,153,151	4,736,897
Avg. Review Length	117.93	115.85
#Avg. Friends	29.87	33.67

Table 1: Statistics of Yelp Round 9 and Yelp Round 10 datasets.

4 Experiments

In this section, we show experiments to demonstrate the effectiveness of random walk based social regularizations for word embeddings.

4.1 Datasets

We conducted all experiments on Yelp Challenge[1] datasets which provide a lot of review texts along with large social networks. At Yelp, people can write reviews for restaurants, bars, etc., and can follow other users to share information. From the simple statistics shown in Table 1, we can see Yelp Round 10 has more reviews and users than Yelp Round 9.

4.2 Experimental Settings

We randomly split data to be 8:1:1 for training, developing, and testing identically for both training word embeddings and downstream tasks, in which we ensure that reviews published by the same user can be distributed to training, development, and test sets according to the proportion. All the following results are based on this fixed segmentation. For SWE, we use the code released by [Zeng *et al.*, 2017].

We use CBOW [Mikolov *et al.*, 2013a] to train word embeddings for all methods. To make a fair comparison, we set the hyper-parameters to be the same as the SWE. For constraint r_1, we empirically set it to $r_1^2 = 0.2r_2^2$.

The social regularization using first-order random walks (**SR-FRW**), using second-order random walks (**SR-SRW**), and using biased (second-order) random walks (**SR-BRW**) involve extra hyper-parameters and they have some hyper-parameters in common. We train embeddings on the training set and search hyper-parameters on the development set using the sentiment classification task. We use the following strategy to reduce time on searching. We perform grid search for SR-FRW, SR-SRW, and SR-BRW in sequence to determine optimal hyper-parameters. Once one method performs grid search on some hyper-parameters and get optimal values, other methods will not search them again. Hence, we will not

[1] https://www.yelp.com/dataset_challenge

Dataset	Yelp Round 9	Yelp Round 10
#Users	16,768	18,976
#Avg. Reviews	11.68	11.63
#Avg. Friends	27.83	27.48

Table 2: Statistics of one-fifth of Yelp Round 9 and 10 data.

Dataset	Yelp Round 9		Yelp Round 10	
	Dev	Test	Dev	Test
W2V	58.98	58.90	59.79	60.09
SWE	59.28	59.12	60.11	60.31
SR-SRW	59.28	59.32	60.24	60.45
SR-BRW	59.28	**59.44**	60.32	**60.53**

Table 3: Sentiment classification accuracies (in %) on one-fifth development and test sets.

report results of SR-FRW since SR-SRW will outperform or be the same as SR-FRW. Finally, we set $\beta = 0.5$ in Yelp Round 9, $\beta = 1.0$ in Yelp Round 10, and $l = 60$, $n = 10$, $p = 0.5$, $q = 1$, $\alpha = 0.12$, $\lambda = 8.0$, $r_2 = 0.25$ in both datasets. Unless we test the parameter sensitivity of our algorithms, we will fix all the hyper-parameters for the following experiments.

4.3 Sentiment Classification

In this section, we evaluate different regularizations on sentiment classification task for Yelp reviews. As shown in [Yang and Eisenstein, 2017], taking language variance and linguistic homophily into consideration can help sentiment analysis task. For example, words such as "good" can indicate different sentiment ratings depending on the author. Hence, it is a valid task to demonstrate the effectiveness of socialized word embeddings. In Yelp, users can write text reviews to describe his/her feelings and opinions towards businesses and then give a star rating. We take the averaged word embeddings of all words (except stop words) in a review as input, and then use the one-vs-rest logistic regression implemented by LibLinear[2] to predict the ratings scaled from 1 to 5.

We compare our social regularizations with two baseline embedding methods, namely, **W2V** and **SWE**. For efficiency, we randomly select one-fifth of the training data to train a logistic regression classifier. The statistics of one-fifth training data are shown in Table 2. In Table 3, results suggest that SWE has better performance than W2V. Social regularizations using random walks outperform SWE, and SR-BRW performs the best among all social regularizations.

As pointed out by [Zeng *et al.*, 2017], previous studies usually preprocessed the data and applied their methods on partial data containing sufficient user information [Tang *et al.*, 2015; Chen *et al.*, 2016]. Hence, we also report the performance on partial data. For efficiency, we still use the same one-fifth of the training data as our training set. But we perform the same preprocessing steps as [Zeng *et al.*, 2017] to obtain head and tail users' data. Here for head users, we mean users published a lot of reviews, while tail users publish less. The statistics of head and tail users are shown in Table 4.

[2]https://www.csie.ntu.edu.tw/~cjlin/liblinear

Dataset	Yelp Round 9		Yelp Round 10	
	Head	Tail	Head	Tail
#Users	3,631	13,137	4,123	14,853
#Avg. Reviews	26.97	7.45	26.75	7.43
#Avg. Friends	72.59	20.06	72.01	19.99
Perc. in SR-SRW	56.20%	43.80%	56.68%	43.32%
Perc. in SR-BRW	44.03%	55.97%	45.94%	54.06%

Table 4: Statistics of head users and tail users in the one-fifth of the training set. "Perc." means the percentage of head/tail users sampled in the random walk paths.

From the table, we can see that head users tend to publish more reviews and have more friends than tail users.

We conduct experiments using the head and tail subsets as training data respectively. To evaluate the significance of the improvements, we run experiments ten times on randomly sampled 60% of the one-fifth training data to report mean, standard deviation, and t-test results. The results are shown in Table 5. We can see that both random walks based methods outperform SWE on both head and tail data. It reflects that our explicit modeling on transitivity is better. From statistics in Table 4, compared with SR-SRW, the proportion of tail users sampled in SR-BRW increases, which shows that the biased coefficient and restart can help to sample more tail users. Moreover, in Table 5, SR-BRW outperforms SR-SRW, so we can conclude that sampling more tail users can improve performance. It is interesting that improvements in head users are more significant than in tail users. For example, in Yelp 10, SR-BRW improve SWE by 1.0% in head users while 0.4% in tail users. The reason might be that SWE will enforce all one-hop neighbors of a head user to be similar to the head. But in reality, head users might have many friends, e.g., 1,000. Forcing all one-hop neighbors to be similar to the head is unreasonable. In our method, only one-hop neighbors sampled by paths are forced to be similar to the head user, others can still maintain their personal characteristics of language use. But tail users do not have this problem.

4.4 Parameters Sensitivity

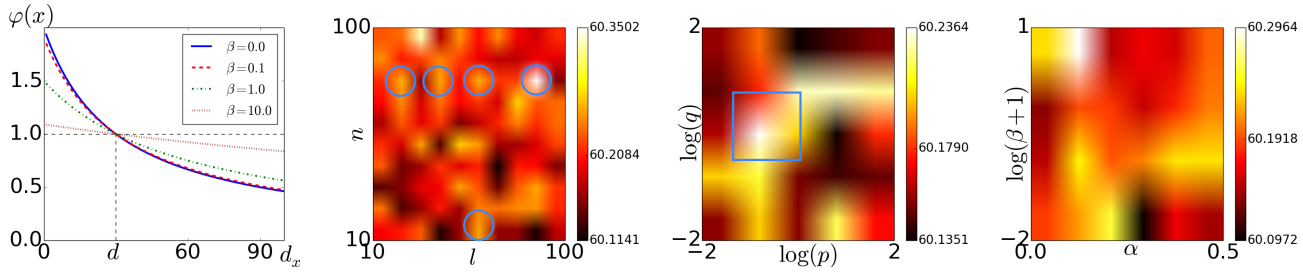
We first perform a grid search over n and l for SR-FRW. In Figure 1(b), areas highlighted by blued circles represent satisfying accuracies. When $n = 80$, accuracies are consistently high, which means l is insensitive when $n = 80$. But we cannot find such a value of l where accuracies always are satisfactory with varying n, which indicates n is more sensitive than l.

Hyper-parameter p and q work together to control exploration strategies, so we perform grid search over p and q for SR-SRW. In Figure 1(c), many light areas are connected, e.g., the area within the blue square, which means we can find a good combination easily in a wide range.

Figure 1(d) shows that light areas are very large, which means it is easy to search satisfying hyper-parameters. When $\alpha = 0.1$ or $\beta = -0.9$, accuracies are consistently high, which means when we fix one hyper-parameter to a certain value, the other one can be insensitive. Compare to $\alpha = 0$, the accuracies of $\alpha = 0.1$ are consistently better, which indicates the restart is effective as long as α falls in the right range.

Dataset	Yelp Round 9			Yelp Round 10		
	Overall	Head	Tail	Overall	Head	Tail
W2V	58.90 (0.03)	56.51 (0.16)	59.32 (0.08)	60.09 (0.03)	57.93 (0.10)	60.41 (0.05)
SWE	59.13 (0.04)	57.99 (0.14)	59.54 (0.07)	60.25 (0.03)	59.42 (0.10)	60.55 (0.05)
SR-SRW	59.38* (0.03)	59.18* (0.12)	59.69* (0.06)	60.46* (0.03)	60.35* (0.07)	60.82* (0.03)
SR-BRW	**59.43*** (0.03)	**59.28*** (0.07)	**59.72*** (0.06)	**60.52*** (0.03)	**60.47*** (0.05)	**60.90*** (0.04)

Table 5: Mean and standard deviation of accuracies (in %) on full one-fifth test data. Overall / Head / Tail means training on randomly sampled 60% full one-fifth / only head users' / only tail users' data. The marker * refers to p-value < 0.0001 in t-test compared with SWE.



(a) $\varphi(x)$ varies with β (b) n and l (c) p and q (d) α and β

Figure 1: Illustration of hyper-parameters in random walks. Heat maps (b - d) show performance under different combinations of parameters. Lighter color means a higher accuracy.

Dataset	Yelp Round 9				Yelp Round 10			
Method	**HCNN**		**HLSTM**		**HCNN**		**HLSTM**	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
W2V without attention	65.28	65.22	66.85	66.98	66.27	66.19	67.80	67.69
W2V with trained attention	65.89	65.97	66.93	66.71	67.04	66.76	67.96	67.61
SWE fixed user vectors as attention	66.31	66.39	66.99	66.75	67.14	66.93	68.21	67.96
SR-SRW fixed user vectors as attention	66.33	**66.43**	67.35	**67.14**	67.19	**67.07**	68.23	68.01
SR-BRW fixed user vectors as attention	66.33	66.33	67.28	67.12	67.28	67.00	68.27	**68.08**

Table 6: Comparison of our model and other baseline methods in accuracy (%) on user attention based deep learning for sentiment analysis.

4.5 User Vectors for Attention

For document-level sentiment classification on Yelp data, the most interesting work [Chen *et al.*, 2016] shows that by using a user attention vector, accuracies can be improved comparing to original models. This way is consistent with SWE framework because it embeds not only words but also users. It is natural to adopt user vectors from methods under SWE framework (SWEs), i.e., SWE, SR-SRW, and SR-BRW, as user attention vectors.

We design three settings in experiments to demonstrate the effectiveness, namely, without user attention, using the user vector from SWEs as fixed attention, and trainable attention. From the perspective of learning a user attention, fixed attention is an unsupervised method since it is trained by one of the methods under the SWE framework while trainable attention is a supervised method since training an attention vector requires supervision of rating scores.

We compare three settings using hierarchical convolutional neural networks (HCNN) and hierarchical long short term memory recurrent neural networks (HLSTM) [Tang *et al.*, 2015; Chen *et al.*, 2016]. We follow the same settings as [Zeng *et al.*, 2017]. For without attention and trainable attention, we use word embeddings from W2V as input. For fixed attention, we use global embeddings from SWEs as input. We train on same one-fifth training data of Yelp Round 9 and Round 10 and evaluate on the same one-fifth development and test sets.

In Table 6, these unsupervised methods even outperform the supervised one, which demonstrates the effectiveness of user vectors under SWE framework. Although fixed attention is unsupervised, it uses rich information from social network while trained attention does not, which might explain the good performance of unsupervised methods. We believe that user and word embeddings trained by unsupervised methods can enhance performance in social network related text understanding tasks if mining rich social information. Moreover, random walk based methods outperform SWE, which demonstrates the effectiveness of explicit modeling of transitivity. It is interesting that in this experiment SR-BRW could be a little worse than as SR-SRW. This may be because in deep learning framework, the other parameters can be trained based on the fixed user embeddings. SR-SRW without bias may keep more fidelity to the user graph, which results in better training for the other parameters. On the contrary, SR-BRW modified the sampling process in the random walk, which may be better for linear logistic regression since it uses averaged word embeddings as input which is less flexible to learn the parameters.

5 Conclusion

In this paper, we adopt random walk based social regularizations to explicitly model the transitivity of language use. Moreover, we propose a biased random walk based social regularization to sample more users who have fewer friends. We demonstrate the effectiveness of our random walk based social regularizations. One important future work would be how to use more social information, not just the number of friends, to improve socialized word embeddings.

Acknowledgements

This paper was supported by China 973 Fundamental R&D Program (No. 2014CB340304) and the Early Career Scheme (ECS, No. 26206717) from Research Grants Council in Hong Kong. Ziqian Zeng has been supported by the Hong Kong Ph.D. Fellowship. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan GPU used for this research.

References

[Chen *et al.*, 2016] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. Neural sentiment classification with user and product attention. In *EMNLP*, pages 1650–1659, 2016.

[Croft *et al.*, 2001] W Bruce Croft, Stephen Cronen-Townsend, and Victor Lavrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, volume 3, page 13, 2001.

[Eckert and McConnell-Ginet, 2003] Penelope Eckert and Sally McConnell-Ginet. *Language and gender*. Cambridge University Press, 2003.

[Fischer, 1958] John L Fischer. Social influences on the choice of a linguistic variant. *Word*, 14(1):47–56, 1958.

[Green, 2002] Lisa J Green. *African American English: a linguistic introduction*. Cambridge University Press, 2002.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[Harris, 1954] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[Hovy, 2015] Dirk Hovy. Demographic factors improve classification performance. In *ACL (1)*, pages 752–762, 2015.

[Huang *et al.*, 2014] Yu-Yang Huang, Rui Yan, Tsung-Ting Kuo, and Shou-De Lin. Enriching cold start personalized language model using social network information. In *ACL*, volume 2, pages 611–617, 2014.

[Labov, 1963] William Labov. The social motivation of a sound change. *Word*, 19(3):273–309, 1963.

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

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Pan *et al.*, 2004] Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, and Pinar Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD*, pages 653–658, 2004.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710, 2014.

[Rosenthal and McKeown, 2011] Sara Rosenthal and Kathleen McKeown. Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations. In *ACL*, pages 763–772, 2011.

[Song *et al.*, 2010] Wei Song, Yu Zhang, Ting Liu, and Sheng Li. Bridging topic modeling and personalized search. In *COLING: Posters*, pages 1167–1175, 2010.

[Sontag *et al.*, 2012] David Sontag, Kevyn Collins-Thompson, Paul N Bennett, Ryen W White, Susan Dumais, and Bodo Billerbeck. Probabilistic models for personalizing web search. In *WSDM*, pages 433–442, 2012.

[Tang *et al.*, 2015] Duyu Tang, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, pages 1014–1023, 2015.

[Trudgill, 1974] Peter Trudgill. Linguistic change and diffusion: Description and explanation in sociolinguistic dialect geography. *Language in society*, 3(2):215–246, 1974.

[Vosecky *et al.*, 2014] Jan Vosecky, Kenneth Wai-Ting Leung, and Wilfred Ng. Collaborative personalized twitter search with topic-language models. In *SIGIR*, pages 53–62, 2014.

[Weiss, 1983] George H Weiss. Random walks and their applications: Widely used as mathematical models, random walks play an important role in several areas of physics, chemistry, and biology. *American Scientist*, 71(1):65–71, 1983.

[Yan *et al.*, 2016] Rui Yan, Cheng-Te Li, Hsun-Ping Hsieh, Po Hu, Xiaohua Hu, and Tingting He. Socialized language model smoothing via bi-directional influence propagation on social networks. In *WWW*, pages 1395–1406, 2016.

[Yang and Eisenstein, 2017] Yi Yang and Jacob Eisenstein. Overcoming language variation in sentiment analysis with social attention. *TACL*, 5:295–307, 2017.

[Zeng *et al.*, 2017] Ziqian Zeng, Yichun Yin, Yangqiu Song, and Ming Zhang. Socialized word embeddings. In *IJCAI*, pages 3915–3921, 2017.