User Identity Linkage Across Social Networks Based on Neural Tensor Network

Xiaoyu Guo[1], Yan Liu[1(✉)], Xianmin Meng[2], and Lian Liu[2]

[1] PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China
hello_dreamer@126.com, ms.liuyan@foxmail.com
[2] Investigation Technology Center PLCMC, Beijing 100000, China
minmax-007@163.com, lianll1024@hotmail.com

Abstract. User Identity Linkage (UIL) across social networks refers to the recognition of the accounts belonging to the same individual among multiple social network platforms. The most existing methods usually apply network embedding to map the network structure space to the low-dimensional vector space and then use linear models or standard neural network layers to measure the correlations between users across social networks. However, they can hardly model the complicated interactions between users. In this paper, we propose a novel Neural Tensor Network-based model for UIL, called NUIL. Firstly, we use the Random Walks and Skip-gram model to learn the vector representations of users. Then, we apply the Neural Tensor Network, which has a stronger ability to express the interactions between entities, to mine relationships between users from a higher dimension. A series of experiments conducted on a real-world dataset show that NUIL outperforms the state-of-the-art network structure-based methods in terms of precision, recall, and F1-measure, specifically the F1-measure exceeds 0.66, with an increase of more than 20%.

Keywords: User identity linkage · Neural tensor network · Network embedding · Social network analysis

1 Introduction

With the rapid development of online social network, people usually join multiple social networks simultaneously according to their needs of work or life [1]. Each user often has multiple separate accounts in different social networks. However, these accounts belonging to the same user are mostly isolated without any connection or correspondence to each other [2].

The typical aim of User Identity Linkage (UIL) is to detect that users from different social platforms are actually one and the same individual, also known as Account Identification, Anchor Link Prediction, and Network Alignment [3]. UIL plays an important role in social network analysis, such as user behavior prediction, friend recommendation across platforms, and information dissemination across networks, etc.

D. Wang et al. (Eds.): SPNCE 2020, LNICST 344, pp. 162–171, 2021.
https://doi.org/10.1007/978-3-030-66922-5_11

Early research uses the public attributes and statistical features of users to solve the UIL problem [4, 5], such as username, user's hobbies, language patterns, etc. However, the correctness and richness of user's public attributes cannot be guaranteed. Compared with user's attributes, the connections between users are reliable and rich, and can also be directly used to solve the UIL problem. Therefore, the methods based on network structure are receiving more and more attention [6, 7].

With the development of network embedding (NE), many people use NE instead of traditional feature engineering to save the structural features of social network into low-dimensional vector space, which not only reduces the complexity of the algorithm, but also improves the accuracy of user identity linkage. For example, Man et al. [8] employed network embedding to capture the major and specific structural regularities and further learned a stable cross-network mapping for predicting anchor links. Zhou et al. [9] propose the FRUIP model which extracts the friend feature vector from the network neighborhood patterns and establishes a "one-to-one" mapping based on the similarity between users.

The existing methods generally use a linear model or standard neural network layers to measure the correlations between users after obtaining the low-dimensional vector space of social networks. However, the relationships between users across social networks are extremely complex, traditional methods can hardly model the complicated interactions between them. Inspired by the success of neural tensor network (NTN) for explicitly modeling multiple interactions of relational data [10], we propose a novel model NUIL: Neural Tensor Network for User Identity Linkage across Social Networks. The contributions of this manuscript are as follows:

- NUIL applies the Random Walks model and Skip-gram model to embed the network structure into a low-dimensional vector space to learn the effective vector representations of nodes and we also compare the performance of different Random Walk strategies in solving the UIL problem.
- NUIL replaces a standard neural network model with a neural tensor network model, which has a stronger ability to express the relationships between users, to relate two user vectors across multiple dimensions.
- We conduct a series of experiments on a real-world dataset consisting of two real social networks. The results show that NUIL can significantly improve the precision, recall, and F1-measure of user identity linkage compared to the state-of-the-art methods, e.g., more than 0.66 in terms of F1-measure.

2 Preliminaries

This section describes the terminologies used in this paper and then formally defines the problem of user identity linkage.

2.1 Terminology Definition

We consider a set of social networks as $G^1, G^2, \ldots, G^n$, each of which is represented as an undirected and unweighted graph. Let $G = (V, E)$ represent the network, where V

is the set of nodes, each representing a user, and E is the set of edges, each representing the connection between two users.

In this paper, we take two social networks as an example, which are treated as source network, $G^s = (V^s, E^s)$, and target network, $G^t = \left(V^t, E^t\right)$ respectively. For convenience, we have the following definitions.

Definition 1 (Anchor Link). *Link* (v_i^s, v_k^t) *is an anchor link between* G^s *and* G^t *iff.* $\left(v_i^s \in V^s\right) \wedge \left(v_k^t \in V^t\right) \wedge \left(v_i^s \text{ and } v_k^t \text{ are accounts owned by the same user in } G^s \text{ and } G^t \text{ respectively}\right)$.

Definition 2 (Anchor Users). *Users who are involved in two social networks simultaneously are defined as the anchor users (nodes) while the other users are non-anchor users (nodes).*

Definition 3 (Corrupted Anchor Link). *If a user from* G^s *and another user from* G^t *do not belong to the same natural person, we call that the two users form a corrupted anchor link.*

2.2 Problem Definition

Based on the definitions of the above terms, we formally define the problem of user identity linkage across social networks. Supposing we have two social networks, G^s and G^t, the UIL problem is to determine whether a pair of users, (v_i^s, v_k^t), $v_i^s \in V^s$, $v_k^t \in V^t$, corresponds to the same real natural person, which can be formally defined as:

$$\Phi\left(v_i^s, v_k^t\right) = \begin{cases} 1, \ v_i^s = v_k^t, \\ 0, \ otherwise. \end{cases} \tag{1}$$

where $\Phi\left(v_i^s, v_k^t\right) = 1$ means v_i^s and v_k^t belong to the same individual.

3 NUIL: The Proposed Model

The existing methods usually apply network embedding to map the social network structure space to the low-dimensional vector space, and then transform the UIL problem into a binary classification problem using a standard neural network layer, whose ability to express the relations between users is relatively weak. Inspired by the success of neural tensor network for explicitly modeling multiple interactions of relational data, we propose a neural tensor network-based framework for user identity linkage across social networks.

As shown in Fig. 1, the framework consists of three main components: Network Embedding, Neural Tensor Network, and Multi-layer Perceptron. Firstly, we generate multiple social sequences by applying Random Walks to sample the network, and then embed each user with a vector using the Skip-gram model. Secondly, we use the neural tensor network to mine the complicated interactions between users. Thirdly, the Multi-layer Perceptron model is used to transform the UIL problem into a binary classification problem. We will explain our model in detail later.

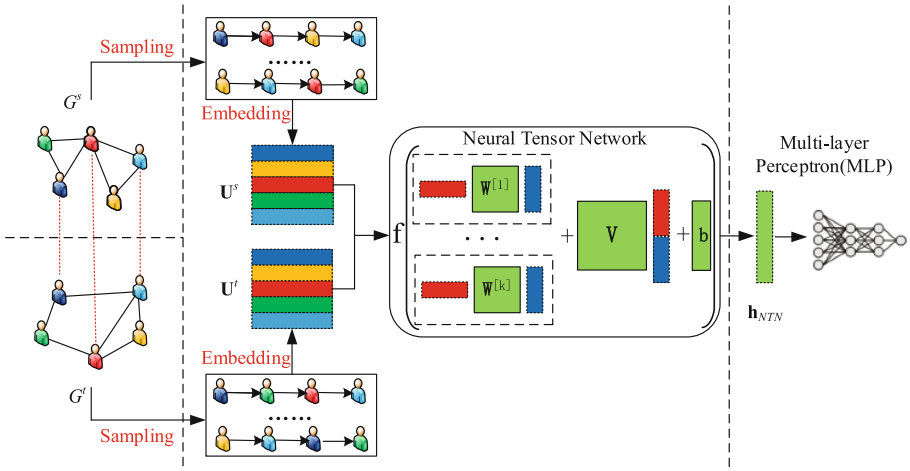

Fig. 1. The framework of NUIL: the red dashed lines represent anchor links. (Color figure online)

3.1 Network Embedding Based on Random Walks and Skip-Gram

To embed users into a latent space, we first generate multiple social sequences for each user in several rounds random walks, which encodes the social relationships among users in the social networks. All the sequences, called "corpus", are used in the Skip-gram model to learn the embedding vectors of users [11].

Network Structure Sampling Based on Random Walks. We conduct the network structure sampling as follows, taking the source network as an example. It starts at node v_m^s and proceeds along a randomly selected edge at each step, until length L is reached. The node sequence of v_m^s is written as $S_{v_m^s}^r$, where r is the rounds of sampling. Sampling by random walks can extract hidden structural social information, e.g., friendship and community in the social network.

User Latent Space Embedding. After getting the "corpus", we apply the Skip-gram model to generate the vector representations of nodes. Skip-gram model is originally used to predict the context of a word by maximizing the average log probability in the domain of word representation. We formally define the node sequence as $v_1, v_2, \ldots, v_L$, and maximize the log probability by the following equation:

$$\frac{1}{L}\sum_{t=1}^{L}\sum_{j=-w}^{w}\log p(v_{t+j}|v_t), j \neq 0 \tag{2}$$

where w is the size of the sliding window and L represents the length of the node sequence.

The conditional probability $p(v_{t+j}|v_t)$ is defined by a SoftMax function, which means the occurrence of the *j-hop* neighbor v_{t+j} given user v_t:

$$p\left(v_{t+j}|v_t\right) = \frac{\exp\left(\boldsymbol{u}_{t+j}^T \boldsymbol{u}_t^{'}\right)}{\sum_{i=1}^{L}\exp\left(\boldsymbol{u}_i^T \boldsymbol{u}_t^{'}\right)} \tag{3}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{u}_i^{'}$ are, respectively, the input and output vector representations of user v_i.

But for a large-scale network, the calculation of $\sum_{i=1}^{L} \exp\left(\boldsymbol{u}_i^T \boldsymbol{u}_t^{'}\right)$ is expensive. Therefore, the Negative Sampling [12] is adopted. The objective function is approximately converted as:

$$\log\left[\sigma\left(\boldsymbol{u}_{t+j}^T \boldsymbol{u}_t^{'}\right)\right] + \sum_{i=1}^{K} \mathbb{E}_{v_i \sim p_n(v)}\left[\log\left(1 - \sigma\left(\boldsymbol{u}_i^T \boldsymbol{u}_t^{'}\right)\right)\right] \tag{4}$$

where K is the number of negative examples. Empirically, each node is sampled with probability $p_n(v) \sim d_{v_i}^{3/4}$, where d_{v_i} is the degree of node v_i [13].

The objective function (2) is approximated by maximizing the objective function (4), and a vector representation of each node v_i is obtained by training using a stochastic gradient descent algorithm.

We apply the network embedding on the source and the target network respectively to obtain the corresponding vector spaces $\mathbf{U}^s$ and $\mathbf{U}^t$.

3.2 Modeling Relations Between Users Based on Neural Tensor Network

Neural Tensor Network. In deep learning literature, neural tensor network is originally proposed to reason the relationships between two entities in knowledge graph or used to classify relation between two entities [10]. It replaces a standard linear neural network layer with a bilinear tensor layer which relates two entity vectors across multiple dimensions.

Given two entities $(\mathbf{e}_1, \mathbf{e}_2)$ represented with d dimensional features, the goal of NTN is to predict whether they have a certain relationship R. Specifically, NTN computes a score of how likely it is that these two entities are in certain relationship R by the following function:

$$g(\mathbf{e}_1, R, \mathbf{e}_2) = \boldsymbol{\mu}_R^T \tanh\left(\mathbf{e}_1^T \mathbf{W}_R^{[1:k]} \mathbf{e}_2 + \mathbf{V}_R \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix} + \mathbf{b}_R\right) \tag{5}$$

where $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^d$ are the vector representations of two entities, $\mathbf{W}_R^{[1:k]} \in \mathbb{R}^{d \times d \times k}$ is a tensor and the bilinear tensor product $\mathbf{e}_1^T \mathbf{W}_R^{[1:k]} \mathbf{e}_2$ results in a vector $\mathbf{h} \in \mathbb{R}^k$, where each entry of $\mathbf{h}$ is computed by one slice $i (i = 1, \cdots, k)$ of the tensor: $h_i = \mathbf{e}_1^T \mathbf{W}_R^{[i]} \mathbf{e}_2$. The other parameters for relation R are the standard form of a neural network: $\mathbf{V}_R = \mathbb{R}^{k \times 2d}$ and $\mathbf{b}_R \in \mathbb{R}^k$. $\boldsymbol{\mu}_R^T \in \mathbb{R}^k$ is used to convert the output of layer to a scalar as a score of the pair of entities for the specific relation, which is high when the entities contain the relation.

Modeling Relations Between Identities in NUIL. The neural tensor network models the relationships between two entities with a bilinear tensor product. We can naturally expand this idea: modeling relations between users, one from the source network G^s and another from the target network G^t, based on the NTN model.

Specifically, for any pair of users $\left(v_m^s, v_n^t\right)$, $v_m^s \in G^s, v_n^t \in G^t$, we model the relationship between them according to the following equation:

$$\mathbf{h}_{NTN}\left(\boldsymbol{u}_m^s, \boldsymbol{u}_n^t\right) = f\left((\boldsymbol{u}_m^s)^T \mathbf{W}^{[1:k]} \boldsymbol{u}_n^t + \mathbf{V}\begin{bmatrix} \boldsymbol{u}_m^s \\ \boldsymbol{u}_n^t \end{bmatrix} + \mathbf{b} \right) \tag{6}$$

where $\begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$ is the concatenation operator on two column vectors.

Through the NTN model described above, we represent the relationship between a pair of nodes $\left(v_m^s, v_n^t\right)$ as $\mathbf{h}_{NTN}\left(\boldsymbol{u}_m^s, \boldsymbol{u}_n^t\right)$. The difference from the original NTN model is that after obtaining the vector $\mathbf{h}_{NTN}$, we do not apply $\boldsymbol{\mu}^T$ to convert it to a scalar, but use it as the input of a Multi-Layer Perceptron.

3.3 Matching Identities Based on Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) is also known as Artificial Neural Network (ANN). There can be multiple hidden layers between the input layer and the output layer, and the layers are fully connected. In addition to the input layer, each node is a neuron with a nonlinear activation function.

The basic problem to be solved by neural networks is the classification problem, and the classic neural network model is MLP. In this paper, we apply MLP to transform the problem of node matching across social networks into a classification problem. Specifically, for any pair of nodes $\left(v_m^s, v_n^t\right)$, $v_m^s \in G^s, v_n^t \in G^t$, with the ground-truth label g_{label}, we use NTN structure to model the complex interactions between them as a vector $\mathbf{h}_{NTN}\left(\boldsymbol{u}_m^s, \boldsymbol{u}_n^t\right)$. And then we input it to MLP, and output the predicted label p_{label} to achieve a binary classification.

$$g_{label}\left(v_m^s, v_n^t\right) = \begin{cases} 1, & v_m^s = v_n^t, \\ 0, & otherwise. \end{cases} \tag{7}$$

$$p_{label}\left(v_m^s, v_n^t\right) = MLP\left(\mathbf{h}_{NTN}\left(\boldsymbol{u}_m^s, \boldsymbol{u}_n^t\right)\right) \tag{8}$$

Therefore, combining the formulas (6) (7) (8), we use the cross entropy to construct the loss function of the entire model as:

$$Loss(\boldsymbol{\Omega}, \mathcal{D}, \mathcal{L}) = -\sum\nolimits_{\left(v_m^s, v_n^t\right) \in \mathcal{D}} \left[g_{label} \log p_{label} + (1 - g_{label}) \log(1 - p_{label}) \right] \tag{9}$$

where $\mathcal{D}$ represents the set of node pairs used for model training, and $\mathcal{L}$ is the ground-truth labels corresponding to $\mathcal{D}$. $\boldsymbol{\Omega}$ is the set of parameters in the model. Please note that we abbreviate $g_{label}\left(v_m^s, v_n^t\right)$ and $p_{label}\left(v_m^s, v_n^t\right)$ as g_{label} and p_{label} respectively.

Assuming that the set of known anchor links is M, we construct the node pairs according to the ratio of positive and negative samples as $1 : C$. We apply the back-propagation algorithm and stochastic gradient descent algorithm to train the NUIL model in a supervised way. Finally, we can get a complete model for user identity linkage across social networks.

4 Experiments

In this section, we compare the proposed NUIL model with several state-of-the-art models on a real-world dataset consisting of two real social networks.

4.1 Dataset, Baselines and Parameter Setup, and Evaluation Metrics

Dataset. The real-world dataset is provided by [7], and it contains two social networks, Twitter and Foursquare. Table 1 summarizes the statistics of this dataset.

Table 1. Statistics of twitter-foursquare dataset.

Networks	#Users	#Relations	#Anchor users
Twitter	5120	164919	1609
Foursquare	5313	76792	

The number of anchor links is 1609, which can be seen as positive instances. Firstly, We set the ratio between positive instances (*Anchor Links*) and negative instances (*Corrupted Anchor Links*) to 1: 1. That is to say, for each anchor user v_i^s in Twitter, we randomly select one user from Foursquare, which is not corresponding to v_i^s, to construct a negative instance. Secondly, we set the ratio between training set, validation set, and test set to 8: 1: 1.

Baselines and Parameter Setup. The model we proposed in this paper is based on network structure, so we compare NUIL with several structure-based methods for UIL.

- **PALE:** Predicting Anchor Links via Embedding (PALE) [8] employs network embedding (such as DeepWalk) with awareness of observed anchor links as supervised information to capture the major and specific structural regularities and further learns a stable cross-network mapping for predicting anchor links.
- **FRUIP:** Structure Based User Identification across Social Networks [9], considers friends relationship with unsupervised learning. First, the friend feature vector is extracted with the network neighborhood patterns, and then compute their similarities for linkage prediction.

Parameter Setup. For the NUIL model proposed in this paper, the vector dimensionality d is 64, the number of tensor layers k in NTN is 8, and we set the MLP with two hidden layers: $32d$ (first hidden layer), $8d$ (second hidden layer) and $1d$ (output layer). The learning rate for training is 0.001 and the batch size is set to 8. The baselines are implemented according to the original papers.

Evaluation Metrics. Precision, Recall, and F1-measure are common metrics to evaluate the performance of a classifier. In this paper, we also evaluate all the methods in terms of Precision, Recall and F1-measure.

4.2 User Identity Linkage Performance

In this section, we present the performance of all methods on twitter-foursquare dataset.

Results Analysis. The Precision, Recall and F1 of all the algorithms are shown in Table 2. Based on the experimental results, we perform the following groups of comparisons to analyze the results.

Table 2. Comparisons of P, R and F1 on Twitter-Foursquare Dataset.

	Precision	Recall	F1
PALE [8]	0.5059	0.5342	0.5197
FRUIP [9]	0.5733	0.5342	0.5531
NUIL	**0.6437**	**0.6956**	**0.6686**

- **PALE - FRUIP**. As can be seen from Table 2, the performance on Precision of the two baseline methods exceeds 0.5. And, we can see that FRUIP, which considers the friend relationship in social networks, performs better than PALE which applies the traditional random walks-based network embedding.
- **PALE - NUIL**. The two methods both apply the traditional random walks-based network embedding. But we can find that NUIL, using the neural tensor model to mine the complicated interactions, is 27% higher than PALE on Precision. We can intuitively see the efficient performance of NTN model in solving UIL problem.
- **FRUIP - NUIL**. From Table 2, we can see that the three evaluation metrics of our method are all above 0.6. Specifically, NUIL is more than 12% higher than FRUIP on Precision.

Parameter Analysis. Through the above experimental results analyses, we can directly see the effectiveness of the NUIL model on user identity linkage across social networks. In this section, we analyze the influence of parameters on the problem of UIL, such as the percentage p of anchor nodes used for training, the vector dimensionality d.

We set the dimensionality to 16, 32, 64, and 128 respectively and the percentage to 0.2, 0.4, 0.6, and 0.8. Figure 2 shows the changes of F1-measure with percentage p and dimensionality d. On the whole, the F1-measure of NUIL model gradually increases and reaches convergence as the vector dimensionality and the percentage of anchor nodes used for training increase.

4.3 Discussions

According to our experiments, we have the following discussions.

- By comparing two pairs of models, PALE-FRUIP and FRUIP-NUIL, we can intuitively find that replacing a standard linear neural network with NTN model is very effective on the problem of user identity linkage across social networks.

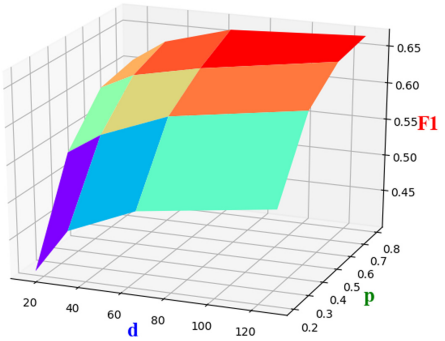

Fig. 2. Parameter Analysis on the vector dimensionality d and the percentage p of anchor nodes used for training.

- The NUIL model can not only be conveniently combined with the current popular network embedding methods, but also can easily be combined with user's attribute information in social networks, such as user's profiles or user's activity patterns.

5 Conclusion

In this paper, we studied the problem of use identity linkage across social networks and proposed a novel model, called NUIL. As the current mainstream methods did, we also applied the network embedding to map the network structure space to the low-dimensional vector space to capture the structural features of social network. Different from the traditional methods with the idea of nodes matching, we introduced the Neural Tensor Network into UIL to convert the node matching problem into a classification problem. The NTN model replaces a standard linear neural network layer with a bilinear tensor layer that directly relates the two entity vectors across multiple dimensions. Several experiments conducted on a real-world dataset indicate the effectiveness of NUIL. In the future work, we are committed to combining more comprehensive network embedding methods with neural tensor network model, such as global structure features of social network, community structure, and user attribute information.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (U1636219, 61602508, 61772549, U1736214, 61572052, U1804263, 61872448) and Plan for Scientific Innovation Talent of Henan Province (No. 2018JR0018).

References

1. Kong, X., Zhang, J., Yu, P.: Inferring anchor links across multiple heterogeneous social networks. In: The 22nd International Conference on Information & Knowledge Management, pp. 179–188. ACM (2013)
2. Shu, K., Wang, S., Tang, J., Zafarani, R., Liu, H.: User identity linkage across online social networks: a review. In: SIGKDD Explorations Newsletter, pp. 5–17. ACM (2017)

3. Zhang, J., Yu, P.: Multiple anonymized social networks alignment. In: 2015 IEEE International Conference on Data Mining, pp. 599–608. IEEE (2015)
4. Goga, O., Lei, H., Parthasarathi, S., Friedland, G., Sommer, R., Teixeira, R.: Exploiting innocuous activity for correlating users across sites. In: The 22nd International Conference on World Wide Web, pp. 447–458. WWW (2013)
5. Zafarani, R., Liu, H.: Connecting users across social media sites: a behavioral-modeling approach. In: The 19th Knowledge Discovery and Data Mining, pp. 41–49. ACM (2013)
6. Wang, C., Zhao, Z., Wang, Y., Qin, D., Luo, X., Qin, T.: DeepMatching: a structural seed identification framework for social network alignment. In: The 38th International Conference on Distributed Computing Systems, pp. 600–610. IEEE (2018)
7. Liu, L., Cheung, W., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: The 25th International Joint Conference on Artificial Intelligence, pp. 1774–1780. IJCAI (2016)
8. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: The 25th International Joint Conference on Artificial Intelligence, pp. 1823–1829. IJCAI (2016)
9. Zhou, X., Liang, X., Du, X., Zhao, X.: Structure based user identification across social networks. IEEE Trans. Knowl. Data Eng. **30**(6), 1178–1191 (2018)
10. Socher, R., Chen, D., Manning, C., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, pp. 926–934. NIPS (2013)
11. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: The 20th Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)
12. Mnih, A., Teh, Y.: A fast and simple algorithm for training neural probabilistic language models. In: The 29th International Conference on Machine Learning, pp. 1751–1758. ICML (2012)
13. Mikolov, T., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119. NIPS (2013)