Hierarchical Attention Based Semi-supervised Network Representation Learning

Jie Liu(✉), Junyi Deng, Guanghui Xu, and Zhicheng He

College of Computer and Control Engineering, Nankai University, Tianjin, China
jliu@nankai.edu.cn, {dengjunyi,xugh,hezhicheng}@mail.nankai.edu.cn

Abstract. Network Embedding is a process of learning low-dimensional representation vectors of nodes by comprehensively utilizing network characteristics. Besides structure properties, information networks also contain rich external information, such as texts and labels. However, most of the traditional learning methods do not consider this kind of information comprehensively, which leads to the lack of semantics of embeddings. In this paper, we propose a Semi-supervised Hierarchical Attention Network Embedding method, named as SHANE, which can incorporate external information in a semi-supervised manner. First, a hierarchical attention network is used to learn the text-based embeddings according to the content of nodes. Then, the text-based embeddings and the structure-based embeddings are integrated in a closed interaction way. After that, we further introduce the label information of nodes into the embedding learning, which can promote the nodes with the same label closed in the embedding space. Extensive experiments of link prediction and node classification are conducted on two real-world datasets, and the results demonstrate that our method outperforms other comparison methods in all cases.

Keywords: Network representation learning
Hierarchical attention network · Semi-supervised learning

1 Introduction

Information network is a data form with rich structure and semantic information. With the prevalence of various social media, massive social networks have attracted a lot of researchers' attention. The applications of information network include various aspects, such as node classification, community detection, and content recommendation. Network representation learning is the foundation of these network applications. Different from the one-hot vectors, network representation can map each node into a low-dimensional, dense and real-valued vector, thus avoiding the effect of sparsity.

Most of the studies on network representation are based on network structure information, such as the sequences generated by network nodes [5,14], the

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 237–249, 2018.
https://doi.org/10.1007/978-3-319-99495-6_20

first-order and second-order proximities [18], and the adjacency matrix [4]. With further research, the external information of nodes are considered to improve the quality of embeddings, such as text information [9,17,19,22] and label information [10,20]. The introduction of text feature can enrich the semantics of nodes and improve the performance of representation learning. It is noteworthy that people usually write sentences first, and then compose the whole document with multiple sentences. However, when considering the text information of a network, existing works usually obtain the text feature matrix of the nodes based on words, which ignores the hierarchical structure. In order to incorporate the document structure (document consists of sentences, sentences consist of words), it is necessary to obtain document representations in a hierarchical way. Besides, different words and sentences contain varying amounts of information, even the same words in different sentences can play different roles. So how to make a difference between different components of nodes' content is a practical problem which needs to be solved. In addition to the text information, label is another important attribute of network nodes, and it is a kind of significant supervised information on directing practical tasks such as classification. Making full use of this supervised information will further enrich the network embeddings [10,20]. However, since the network is usually large-scale, there are still a lot of unlabeled nodes, thus the rational use of labeled data and unlabeled data is important for network representation learning.

In view of the above problems, we propose a hierarchical structure based semi-supervised network representation learning method, **S**emi-supervised **H**ierarchical **A**ttention **N**etwork **E**mbedding (SHANE), which can learn the hierarchical relational network embeddings by integrating text and label features of nodes. In SHANE, we adopt a hierarchical attention structure to extract text features at different levels [23], which can model the hierarchical semantic information of network. Meanwhile, label information is utilized in a semi-supervised manner to make full use of both labeled data and unlabeled data. We apply the proposed model to link prediction and node classification. The experiment results show that the proposed model outperforms all the comparison methods. Our contributions are summarized as follows:

- We propose a SHANE model, which can integrate structures, texts, and labels of nodes together, and learn network embeddings in a semi-supervised manner.
- We use hierarchical attention network to model the nodes' text features, which can capture the semantic features more granularly.
- We extensively evaluate our representations with multiple tasks on two real-world citation networks. Experimental results prove the effectiveness of the proposed model.

2 Model

The overall architecture of the proposed model is shown in Fig. 1. It consists of Word Encoder, Sentence Encoder, and Node Encoder. Word Encoder and Sentence Encoder constitute the text-based representation learning process, while

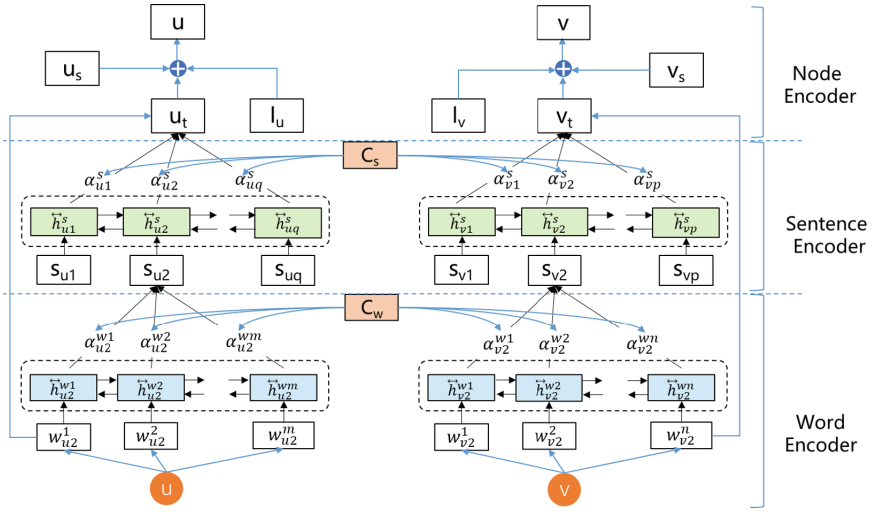

Fig. 1. The illustration of SHANE model.

the Node Encoder combines structure-based embedding, text-based embedding, and label information together. We describe the details of different components in the following sections.

2.1 Problem Formulation

First of all, we introduce the related notions and define the problem formally. Let $G = (V, E, T, L)$ denotes a given information network, where V is the set of nodes, E is the edge set that indicates the relation between nodes, T denotes the text information of nodes and L is the label information of nodes. Each edge $e_{u,v} \in E$ represents the relationship between two nodes (u, v). The text information of node u is $D_u = (S_{u1}, S_{u2}, \cdots, S_{uq})$, where S_{ui} is the ith sentence of u and q is the sentences number of u. $S_{ui} = (w^1_{ui}, w^2_{ui}, \cdots, w^m_{ui})$, where w^j_{ui} is the jth word of sentence S_{ui} and m is the words number of sentence S_{ui}. The label information of u is l_u.

Given an information network, the goal of our model is to learn a low-dimensional vector $\mathbf{u}$ for each node u, that can integrate its structure, text and label information.

2.2 Text-Based Representation Learning

As mentioned above, the text information of nodes usually has a natural hierarchical structure. That is, each document contains multiple sentences, and each sentence contains multiple words. Empirically, each word and sentence are of different importance in a document, and learning all sentences and words indiscriminately will lose the focus of text content. So we use a hierarchical attention

network [23] to learn the text-based embedding $\mathbf{u}_t$ for each node u, and we describe the learning process in details as follow.

Word Encoder. Assume that u contains q sentences, and each sentence contains m words. We can get the word sequence of sentence S_{ui} by table looking-up, so the sentence can be expressed as $S_{ui} = (\mathbf{w}_{ui}^1, \mathbf{w}_{ui}^2, \cdots, \mathbf{w}_{ui}^m)$, where $\mathbf{w}_{ui}^j \in R^d$ is a d-dimensional embedding vector. Then a bidirectional GRU [1] is applied to encode the word sequences as:

$$\begin{aligned} \overrightarrow{\mathbf{h}}_{ui}^{wj} &= \overrightarrow{GRU}(\mathbf{w}_{ui}^j), j \in [1, m], \\ \overleftarrow{\mathbf{h}}_{ui}^{wj} &= \overleftarrow{GRU}(\mathbf{w}_{ui}^j), j \in [m, 1]. \end{aligned} \tag{1}$$

The word annotation $\mathbf{h}_{ui}^{wj}$ of $\mathbf{w}_{ui}^j$ should contain two directions of information, which can be simply obtained by concatenating $\overrightarrow{\mathbf{h}}_{ui}^{wj}$ and $\overleftarrow{\mathbf{h}}_{ui}^{wj}$. Considering that words contribute differently to the sentence representation, the attention mechanism is used to identify the importance of words, and the operations can be expressed as follows:

$$\begin{aligned} \mathbf{g}_{ij} &= \tanh\left(W_w \mathbf{h}_{ui}^{wj} + \mathbf{b}_w\right), \\ \alpha_{ui}^{wj} &= \frac{\exp(\mathbf{g}_{ij}^T C_w)}{\sum(\exp(\mathbf{g}_{ij}^T C_w))}, \\ \mathbf{s}_{ui} &= \sum_j \alpha_{ui}^{wj} \mathbf{h}_{ui}^{wj}, \end{aligned} \tag{2}$$

where $\mathbf{s}_{ui}$ is the embedding of the ith sentence of node u, C_w is the global word-level context vector, and α_{ij} is a normalized importance weight used to fuse word annotations to get the representation of sentence.

Sentence Encoder. Sentence encoder is similar to the word encoder except that the objects are sentences, so we omit the equations due to lack of space. Similar bidirectional GRU and attention layers are applied to the sentence encoding process, and then we can get the text embedding $\mathbf{u}_t^h$ encoded by hierarchical attention network.

To avoid the deviation of the learned representation from the original text, after getting the embedding from hierarchical attention network, we add it with another vector $\mathbf{u}_t^a$, which is the mean of word embeddings of this node. Then, we can get the text-based representation $\mathbf{u}_t$ of node u.

$$\mathbf{u}_t = \mathbf{u}_t^h + \mathbf{u}_t^a. \tag{3}$$

Overall, two layers of bidirectional GRUs extract the latent features of words and sentences, in which the word-level attention is used to capture the lexical features, and the sentence-level attention is used to capture the textual features. Therefore, the hierarchical learning method can obtain text information with different granularities.

2.3 Structure-Based Representation Learning

In addition to the text-based embeddings discussed above, the structures of nodes are also crucial information of the network. Structure features reflect the connection characteristics of nodes. In general, two nodes with an edge between them are similar in structure. Therefore, while getting the text-based embeddings of nodes, we also learn a network structure based embedding u_s for each node. In order to comprehensively learn the node representations, it is necessary to consider the correlation between structure features, the relationship between text features, and their interactions.

Following CANE [19], we set the log-likelihood functions of each part as follows:

$$\begin{aligned} L_{ss}(u) &= \sum_{e_{u,v} \in E} w_{u,v} \log p(\mathbf{v}_s^u \mid \mathbf{u}_s^v), \\ L_{tt}(u) &= \sum_{e_{u,v} \in E} w_{u,v} \log p(\mathbf{v}_t \mid \mathbf{u}_t), \\ L_{st}(u) &= \sum_{e_{u,v} \in E} w_{u,v} \log p(\mathbf{v}_s^u \mid \mathbf{u}_t), \\ L_{ts}(u) &= \sum_{e_{u,v} \in E} w_{u,v} \log p(\mathbf{v}_t \mid \mathbf{u}_s^v), \end{aligned} \tag{4}$$

where v is a node connected with u. $w_{u,v}$ is the weight of the edge between node u and v. $\mathbf{u}_s^v$ is the structure-based embedding of u when it connects with v. The uses of symbols of v are analogous to u. Thus, we can comprehensively model the interaction between u and v through Eq. 4. For each $\mathbf{u} \in \{\mathbf{u}_s^v, \mathbf{u}_t\}$ and $\mathbf{v} \in \{\mathbf{v}_s^u, \mathbf{v}_t\}$, the conditional probability of $\mathbf{v}$ generated by $\mathbf{u}$ is defined through a softmax function:

$$p(\mathbf{v} \mid \mathbf{u}) = \frac{\exp(\mathbf{u}^T \cdot \mathbf{v})}{\sum_{z \in V} \exp(\mathbf{u}^T \cdot \mathbf{z})}. \tag{5}$$

The structure-based embeddings are free parameters to learn, and the text-based embeddings are obtained through the method described in the previous section. Note that the structure-based embeddings of u are different according to the node it connects, and the motivation of this setting is that a node has different connection characteristics when connected with different nodes. The final structure-based embedding is the mean of them:

$$\mathbf{u}_s = \frac{1}{|E_u|} \sum_{e_{u,v} \in E} \mathbf{u}_s^v, \tag{6}$$

where $|E_u|$ is the edges number of u.

2.4 Semi-supervised Hierarchical Attention Network Embedding

Label is another valuable external information of nodes. Nodes with the same label may also be similar in representations. Thus in this section, we incorporate

label information into the learning process. However, the label information of a network in the real world is mostly incomplete, and only a subset of nodes have the corresponding class labels. Therefore, we design our model under a semi-supervised manner so that it can make full use of labeled and unlabeled nodes simultaneously.

Firstly, for the unlabeled nodes, we only consider its structure and text features. So we add the log-likelihood functions in Eq. 4 together to get the objective function of unlabeled nodes:

$$\begin{aligned} L_{unlabel}(u^u) = \alpha \cdot L_{ts}(u^u) + \beta \cdot L_{tt}(u^u) + \theta \cdot L_{st}(u^u) \\ + \gamma \cdot L_{ts}(u^u), \end{aligned} \tag{7}$$

where $u^u \in L_u$ and L_u represents the unlabeled node subset, and $\alpha, \beta, \theta, \gamma$ control the weights of each part.

For the label matching loss of the nodes, we map the node embeddings into the label space by using a fully-connected layer. Then we can get the nodes' predicted label distributions. The purpose of label matching loss is to minimize the distance between predicted label distribution and ground truth distribution.

$$L_{match}(u^l) = -l_u \log p(\widehat{l_u} \mid u^l) + \Omega, \tag{8}$$

where $u^l \in L_l$, and L_l represents the node subset with label information. l_u is the ground truth and $\widehat{l_u}$ is the predicted label distribution. Ω is the regularizing term for the parameters in Eq. 8. Then the objective function of labeled node u^l can be denoted as follow:

$$\begin{aligned} L_{label}(u^l) = \alpha \cdot L_{ts}(u^l) + \beta \cdot L_{tt}(u^l) + \theta \cdot L_{st}(u^l) \\ + \gamma \cdot L_{ts}(u^l) - \lambda L_{match}(u^l), \end{aligned} \tag{9}$$

where λ is the weight of label matching loss. Therefore, the overall objective function of SHANE can be defined as:

$$L = \sum_{u^l \in L_l} L_{label}(u^l) + \sum_{u^u \in L_u} L_{unlabel}(u^u). \tag{10}$$

2.5 Model Optimization

In order to maximize the objective function, we need to calculate the conditional probabilities, which have an expensive computational cost. So we employ the negative sampling technology [12] to reduce the calculation cost. For each $\mathbf{u} \in \{\mathbf{u}_s^v, \mathbf{u}_t\}$ and $\mathbf{v} \in \{\mathbf{v}_s^u, \mathbf{v}_t\}$, the objective functions in Eq. 4 can be transformed as follow:

$$\log \sigma(u^T \cdot v) + \sum_{i=1}^{k} E_{z \sim P(v)}[\log \sigma(u^T \cdot z)], \tag{11}$$

where k is the number of negative samples, and σ is the sigmoid function. We set $P(v) \propto d_v^{3/4}$ as proposed in [12], where d_v is the degree of node v. So in the process of optimization, we replace the corresponding parts of Eq. 10 with the form of Eq. 11. Then, we use Adam to optimize the whole objective function.

3 Experiments

In this section, experiments are performed to verify the effectiveness of the proposed method, including link prediction and node classification.

3.1 Dataset

We conduct our experiments on two citation networks that are commonly used in network representation learning:

- **Cora** is a citation network. We adopt the version processed by CANE [19]. The network contains 2277 machine learning papers in 7 categories and there are 5214 edges between them. The text features of nodes are the abstracts of these papers.
- **DBLP** is a computer science bibliography. It contains 30422 nodes in 4 research areas with the same setting as that of [13], and the edge number is 41206. Abstracts of these papers are treated as text information as well.

3.2 Baseline

To investigate the performance of the proposed model, we compare our model with 7 state-of-the-art methods, including structure-only models, text-only models, structure-text models and structure-text-label models. The details of the comparison methods are described as follow:

- **DeepWalk** [14] is a structure-only model that employs random walk and Skip-Gram [12] to learn the embeddings of nodes.
- **LINE** [18] can learn nodes embeddings of large-scale networks by considering the first-order and second-order proximities and it is a structure-only model.
- **node2vec** [5] is an improved method of DeepWalk with a biased random walk procedure, which only considers the structure information of network.
- **Doc2vec** [8] learns the embeddings of documents by predicting the co-occurrence probability of words, and it is a pure text representation learning method.
- **TADW** [22] is a structure-text model, which learns the structure features and text features of the network by matrix decomposition.
- **CANE** [19] can learn the content aware embeddings of nodes, and it introduce text information into the learning process.
- **TriDNR** [13] is a network representation learning method that considers the structure, text and label information of nodes simultaneously.

3.3 Link Prediction

Link prediction is an important applications of network representation learning. The primary purpose of this task is to predict whether there is an edge between two nodes in the network. In practical applications, it can be used for recommendation tasks, such as book recommendation. We adopt AUC [6] to evaluate the

performance of link prediction. When the AUC is higher than 0.5, it indicates that the similarity of the connected nodes is higher than the unconnected nodes. So higher AUC means better performance. It can be calculated as follows:

$$\text{AUC} = \frac{\sum_{i \in positive} rank_i - \frac{M(1+M)}{2}}{M \times N}, \tag{12}$$

where M and N are the numbers of the connected node pairs and the unconnected node pairs, respectively. In the evaluation process, we calculate the similarities of the node pairs and rank them. The $rank_i$ is the number of correct orders and *positive* is the collection of connected node pairs in the test set.

In order to verify the effectiveness of each model on link prediction task, the models are trained with different proportions of edges in the network. The training proportion of edges ranges from 15% to 95%, and the experimental results on the two datasets are shown in Tables 1 and 2. It is worth noting that HANE is a simplified form of the model proposed in this paper, which is designed to verify the effect of the introduction of label information. HANE doesn't consider the label information of nodes. Besides, since Doc2vec is a

Table 1. Link prediction performance on Cora.

Training edge	15%	25%	35%	45%	55%	65%	75%	85%	95%
DeepWalk	56.0	63.0	70.2	75.5	80.1	85.2	85.3	87.8	90.3
LINE	55.0	58.6	66.4	73.0	77.6	82.8	85.6	88.4	89.3
node2vec	55.9	62.4	66.1	75.0	78.7	81.6	85.9	87.3	88.2
TADW	86.6	88.2	90.2	90.8	90.0	93.0	91.0	93.4	92.7
CANE	86.8	91.5	92.2	93.9	94.6	94.9	95.6	96.6	97.7
TriDNR	83.7	84.7	85.2	85.5	85.8	85.9	86.3	87.2	87.7
HANE	**93.0**	**94.1**	94.8	95.0	95.5	95.8	96.2	97.5	98.3
SHANE	92.7	93.3	**95.1**	**95.7**	**96.0**	**96.5**	**96.9**	**97.8**	**98.5**

Table 2. Link prediction performance on DBLP.

Training edge	15%	25%	35%	45%	55%	65%	75%	85%	95%
DeepWalk	71.2	72.8	74.3	74.5	74.6	75.5	81.4	81.7	82.4
LINE	57.4	60.5	63.2	66.0	66.4	68.8	67.2	68.2	69.9
node2vec	67.0	79.2	84.4	88.1	90.0	91.8	93.2	94.1	95.4
TADW	72.0	78.5	88.5	86.0	87.9	89.2	90.5	91.3	93.4
CANE	91.0	92.2	94.5	94.6	94.8	94.9	95.2	95.7	96.2
TriDNR	86.2	86.0	86.6	86.7	87.2	87.8	88.4	88.6	90.7
HANE	92.4	**93.6**	**95.3**	95.6	**96.3**	**96.5**	96.7	97.0	97.4
SHANE	**92.6**	93.4	**95.3**	**96.1**	**96.3**	96.4	**96.8**	**97.2**	**97.9**

text-only method, so we do not analyze it in this part of the experiment. From Tables 1, 2, we have the following observations:

- Structure-text methods outperform structure-only methods, especially when the reserving proportion of the edge is small. This phenomenon shows that the introduction of text features can significantly improve the quality of embeddings and capture the internal relationship between nodes better.
- The performances of all methods increase with the training ratio of edges, but the methods considering text information are relatively stable. It proves that adequate structure information is conducive to the representation learning, and it also shows that the introduction of text information can supplement the lack of network structure information.
- Either HANE or SHANE performs better than other comparison methods on Core and DBLP, which proves the effectiveness of the hierarchical structure based method proposed in this paper, and our methods can be well adapted to different scales of networks.
- According to the comparison between HANE and SHANE, the introduction of label information achieves a slight improvement in link prediction. It shows that the label information is not the primary factor in the nodes relation learning on these datasets.

3.4 Node Classification

Node classification is also a typical application of network representation learning. While link prediction can evaluate the ability of models to learn the connection characteristics, node classification can verify the ability to capture the group characteristics of nodes. To reduce the influence of the differences between classifiers, we adopt a standard linear SVM on the embeddings learned by all the methods. We use the Macro-F1 score [11] as the evaluation metric, and the higher Macro-F1 means the better performance of classification. In order to study the performance of models under different label completeness, the classification experiment is conducted under the condition of retaining different ratios of labeled nodes. For the unsupervised models, the changes in the ratio of labeled data are reflected in varying the amount of labeled data used for training classifiers. Experimental results on the two datasets are shown in Tables 3 and 4. From these tables, we have following observations:

- Generally speaking, the performances of structure-text models are superior to the text-only method, and both of them perform better than the structure-only methods, which proves that the text information is critical when learning the group characteristics of the nodes.
- Both TriDNR and SHANE introduce the label and text information into the learning process, but the classification performances of SHANE are better than TriDNR. It shows that the way of introducing external information can also affect the performance of network representation learning.

– The experimental results on two datasets show that the proposed model exhibits consistent superior performance to other comparison methods, which proves the effectiveness of the SHANE model. It can efficiently capture the properties of nodes and improve the quality of the network embeddings.

Table 3. Node classification performance on Cora.

Ratio of labeled nodes	10%	30%	50%	70%
DeepWalk	0.446	0.635	0.697	0.733
LINE	0.259	0.299	0.331	0.353
node2vec	0.715	0.761	0.784	0.793
Doc2vec	0.530	0.617	0.654	0.670
TADW	0.413	0.781	0.838	0.852
CANE	0.825	0.861	0.863	0.871
TriDNR	0.655	0.677	0.714	0.744
SHANE	**0.852**	**0.871**	**0.873**	**0.886**

Table 4. Node classification performance on DBLP.

Ratio of labeled nodes	10%	30%	50%	70%
DeepWalk	0.379	0.454	0.459	0.461
LINE	0.328	0.362	0.371	0.372
node2vec	0.448	0.473	0.475	0.476
Doc2vec	0.574	0.598	0.604	0.605
TADW	0.660	0.687	0.697	0.699
CANE	0.801	0.810	0.817	0.822
TriDNR	0.724	0.742	0.747	0.748
SHANE	**0.806**	**0.811**	**0.821**	**0.850**

Figure 2 shows how the performances of the proposed methods change over the proportion of labeled nodes on Cora. The training proportion of the labeled data ranges from 10% to 70%. It can be seen from the figure that the performances of the models are improved when the proportion of the labeled data increases, but SHANE always performs better than HANE. This phenomenon illustrates that the semi-supervised learning method proposed in this paper is effective. The introduction of label information is beneficial to capture the characteristics of nodes, thus improving the quality of network embeddings.

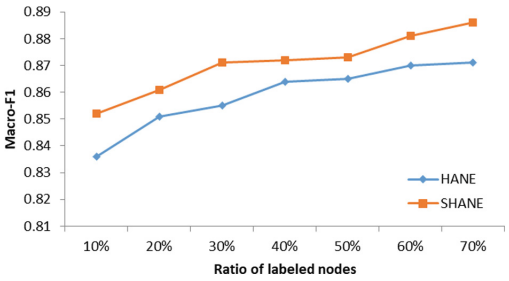

Fig. 2. Performance variation on different training ratio of labeled data.

4 Related Work

Recently, more and more studies focus on how to learn effective network embeddings. The related methods can be divided into two main categories, including the methods only considering network structure, and the methods introducing external information. As a classic structure-only method, DeepWalk [14] learns embeddings of network by performing truncated random walks over networks. On the basis of DeepWalk, Grover et al. proposed node2vec [5] which extends DeepWalk by modifying the random walk strategy. LINE [18] preserves both the first-order proximity and the second-order proximity of the network. These methods can capture the network structure features well, but there is a lack of understanding the semantics of nodes. In order to enrich the semantics of embeddings, many methods introduce external information into the process of learning. PTE [17], CANE [19], and TADW [22] introduce the content of nodes to enrich the network representations. In addition to text, there is also some other external information that can be considered, such as MMDW [20] and DDRW [10]. Although these methods use external information, they do not take into account the hierarchical structure of node content which is an important feature of nodes.

Text representations learning methods are needed when considering the text information of the network. Traditional text representation learning methods, such as LDA [3] and NMF [2], learn the representation of text from the perspective of topic distribution. In recent years, due to the rapid development of neural network and deep learning, text representation learning methods based on the neural network have made significant progress, such as CNNs [7] and LSTM [16] to learn text representations. Recently, attention mechanism is widely used in Natural Language Processing tasks, and Bahdanau et al. first introduced it to NLP for machine translation task [1]. After that, attention mechanism has been widely applied to various applications, such as syntactic parsing [21] and natural language Q&A [15]. The hierarchical attention network [23] proposed by Yang et al. takes into account the hierarchical structure of documents, and applies two levels of attention mechanisms at the word and sentence-level, respectively, which improves the performance of text classification.

5 Conclusion

In this paper, we proposed a semi-supervised hierarchical attention network embedding method, i.e. SHANE. It integrates rich external information into the learning process. The proposed SHANE leverages hierarchical attention network to learn the text-based embedding, which can effectively model the hierarchical structure of the text. Through a semi-supervised learning framework, the embeddings of nodes can be learned by incorporating structure, text and label information together. Extensive experiments conducted on two citation datasets demonstrate the effectiveness and superiority of the proposed model.

Acknowledgement. This research is supported by the National Natural Science Foundation of China under the grant No. U1633103 and 61502499, the Science and Technology Planning Project of Tianjin under the grant No. 17ZXRGGX00170, the Natural Science Foundation of Tianjin under the grant No. 18JCYBJC15800, and the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China under the grant No. CAAC-ITRB-201601.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2014)
2. Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization. Comput. Stat. Data Anal. **52**(1), 155–173 (2007)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Cao, S., Lu, W., Xu, Q.: GraRep: learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 891–900 (2015)
5. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of KDD, pp. 855–864 (2016)
6. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology **143**(1), 29–36 (1982)
7. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of ACL, pp. 655–665 (2014)
8. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning, pp. 1188–1196 (2014)
9. Li, H., Wang, H., Yang, Z., Liu, H.: Effective representing of information network by variational autoencoder. In: Proceedings of IJCAI, pp. 2103–2109 (2017)
10. Li, J., Zhu, J., Zhang, B.: Discriminative deep random walk for network classification. In: Proceedings of ACL (2016)
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS, pp. 3111–3119 (2013)

13. Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y.: Tri-party deep network representation. In: Proceedings of IJCAI, pp. 1895–1901 (2016)
14. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of KDD, pp. 701–710 (2014)
15. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. In: Proceedings of NIPS, pp. 2440–2448 (2015)
16. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of ACL, pp. 1556–1566 (2015)
17. Tang, J., Qu, M., Mei, Q.: PTE: predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1165–1174 (2015)
18. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: Proceedings of WWW, pp. 1067–1077 (2015)
19. Tu, C., Liu, H., Liu, Z., Sun, M.: CANE: context-aware network embedding for relation modeling. In: Proceedings of ACL, pp. 1722–1731 (2017)
20. Tu, C., Zhang, W., Liu, Z., Sun, M.: Max-margin DeepWalk: discriminative learning of network representation. In: Proceedings of IJCAI, pp. 3889–3895 (2016)
21. Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., Hinton, G.E.: Grammar as a foreign language. In: Proceedings of NIPS, pp. 2773–2781 (2015)
22. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E.Y.: Network representation learning with rich text information. In: Proceedings of IJCAI, pp. 2111–2117 (2015)
23. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: Proceedings of the NAACL, pp. 1480–1489 (2016)