Front. Comput. Sci., 2020, 14(6): 146322
https://doi.org/10.1007/s11704-020-8440-6

RESEARCH ARTICLE

Text-enhanced network representation learning

Yu ZHU[1,2,3,4], **Zhonglin YE**[1,2,3], **Haixing ZHAO** (✉)[1,2,3], **Ke ZHANG**[5]

1 School of Computer, Qinghai Normal University, Xining 810008, China
2 Key Laboratory of Tibetan Information Processing and Machine Translation, Xining 810008, China
3 Key Laboratory of Tibetan Information Processing, Ministry of Education, Xining 810008, China
4 Department of Computer Technology and Applications, Qinghai University, Xining 810016, China
5 School of Information Engineering, Huzhou University, Huzhou 313000, China

Abstract Network representation learning called NRL for short aims at embedding various networks into low-dimensional continuous distributed vector spaces. Most existing representation learning methods focus on learning representations purely based on the network topology, i.e., the linkage relationships between network nodes, but the nodes in lots of networks may contain rich text features, which are beneficial to network analysis tasks, such as node classification, link prediction and so on. In this paper, we propose a novel network representation learning model, which is named as Text-Enhanced Network Representation Learning called TENR for short, by introducing text features of the nodes to learn more discriminative network representations, which come from joint learning of both the network topology and text features, and include common influencing factors of both parties. In the experiments, we evaluate our proposed method and other baseline methods on the task of node classification. The experimental results demonstrate that our method outperforms other baseline methods on three real-world datasets.

Keywords network representation, network topology, text features, joint learning

1 Introduction

Modern society has entered an era of information explosion, and information networks are ubiquitous in the real world with examples such as social and communication networks, airline networks, citation networks between academic papers and so on. Recently, in order to extract useful information from massive network data, some researchers have already begun to focus on network representation learning to find a dense, continuous, and low-dimensional vector for each node of the network as its distributed representation, which can be applied to various machine learning tasks, such as node classification [1], recommendation system [2,3], and link prediction [4]. In particular, network representation learning can alleviate the sparse issues caused by the conventional representation methods.

Most related works take the network topology information as input to learn a low-dimensional vector for each node, such as DeepWalk [5], node2vec [6], LINE [7], GraRep [8], SDNE [9] and HARP [10]. These topology-based embedding methods assume that the nodes close in the network topology structure should also be close in the node vector representation space. However, in many real scenarios, the vectors learned purely from the network topology structure are not desirable vectors. For example, in the social network, it is possible that two users with similar interests are not connected and share no common friends, thus the topology-based embedding methods can not effectively capture their similarity on interests. In such a case, other auxiliary information should be incorporated to learn vector representations of better quality.

Usually, the nodes in the network may be associated with a

Received December 21, 2018; accepted January 20, 2020

E-mail: h.x.zhao@163.com

set of features, such as text, community or label information of the nodes, which are useful for measuring the similarity between the nodes. However, most previous works ignore these features. For example, in a paper citation network, if two papers share some abstracts or keywords, they may be similar even if they are topologically far away from each other. Since these node features potentially encode different types of information from the network topology structure, integrating them into the embedding process is expected to achieve a better performance. Based on the above ideas, TADW [11] is proposed to incorporate the node text features into the embedding process under a framework of matrix factorization. However, TADW has the following drawbacks: (1) the very time and memory consuming of matrix factorization process of TADW makes it not scalable to large-scale networks; (2) TADW simply ignores the contexts of text features, so cannot appropriately capture the semantic correlations between the nodes and their related words. CANE [12] is also a recently proposed algorithm to learn vector representations of the nodes from different contexts related to the nodes. The global vector for the node is the concatenation of two types of local vectors: structure-based embedding and text-based embedding. However, CANE fails to consider the inter-relation between structure-based embedding and text-based embedding, which are learned independently. Besides, CANE is not general since it cannot be used when rich text information is available on edges.

In this paper, we propose a general network embedding framework which can effectively encode both the network topology and rich text features of the nodes. There is the following challenge in handling this task. It is not easy how best to combine the network topology and text features of the nodes into a unified embedding process under a general framework. There are sophisticated interactions between the network topology and text features of the nodes, and it is difficult to incorporate text features into the existing topology-based models.

To address the above challenge, we propose a general text-enhanced network representation learning model TENR. We formulate the learning process of text-enhanced network representation as a joint problem, where Topology-Derived model and Text-Derived model are optimized jointly. Specifically, we propose a negative sampling strategy to capture the topology information, which aims to exploit inter-node relationships by maximizing the probability of predicting the node given its contextual nodes in random walks generated from the network. Besides, the negative sampling strategy is still adopted to capture node-text semantic correlations by maximizing the probability of predicting the node given its related words. Finally, we utilize stochastic gradient ascent (SGA) to solve this joint optimization problem.

The contributions of this paper can be summarized as follows:

- We propose a novel network embedding model that captures both the network topology and textual contents. Experiments on the task of node classification using three real-world datasets demonstrate its superiority over various baseline methods.
- We utilize textual contents in the homogeneous network through converting into a heterogeneous network. Homogeneous network and textual contents are integrated into a heterogeneous network, giving us the possibility to integrate and exploit different information.

We test our method against several baseline methods on three datasets. The Micro-F1 and Macro-F1 values for node classification of our method outperform the values of other baselines when the ratios of training sets range from 10% to 90%. Meanwhile, our method shows strong clustering abilities by node clustering visualizations on Citeseer, DBLP and Weibo.

The rest of this paper is organized as follows. Section 2 summarizes the related works. Section 3 gives the formal definition of our studied problem. Section 4 introduces Topology-Derived model, Text-Derived model, TENR model and Complexity analysis in turn. The datasets and experimental results are introduced in Section 5. Section 6 concludes this paper.

2 Related works

Network representation learning aims to learn a distributed vector for each node in a network, which becomes more and more popular in lots of network analysis tasks.

In recent years, there have been lots of NRL models to learn efficient vector representations of the nodes in the network. For example, DeepWalk introduces Skip-Gram [13], a widely-used distributed word representation method, into the study of the network to learn a low-dimensional vector for each node. node2vec modifies the random walk strategy in DeepWalk into biased random walks to explore the network topology structure. LINE optimizes a carefully designed objective function which preserves both the global and local structure. GraRep, with the k-step loss functions defined on graphs which integrate rich local structural information associated with the graph, captures the global structural properties of the graph. As a deep embedding model, SDNE captures the highly non-linear network structure and exploits the first-order proximity and second-order proximity to characterize the local and global network structures. HARP is proposed to learn low-dimensional representations of a graph's nodes to preserve higher-order structural features by compressing the input graph prior to embedding it. Nevertheless, most of these NRL models only encode the topology information into vector representations, without considering other features associated with the nodes in the network, such as text, community or label information of the nodes.

To cope with this issue, researchers make great efforts to incorporate these related features of the nodes into the topology-based models. For example, TADW improves matrix factorization based DeepWalk with text information. MMDW [14] utilizes label information of the nodes to learn discriminative vector representations. CANE learns context-aware embeddings for the nodes with mutual attention mechanism and models the semantic relationships between the nodes. CNRL [15] simultaneously detects community distribution of each node and learns embeddings of both nodes and communities. PPNE [16] incorporates rich types of node properties into the network embedding process. MVC-DNE [17] incorporates both the network structure and the node properties and efficiently performs network embedding on incomplete networks. CENE [18] utilizes both structural and textural information to learn network embeddings. TriDNR [19] utilizes information from three parties: node structure, node content, and node labels (if available) to jointly learn optimal node vector representations. Rank2vec [20] considers both local structure and global structural roles to enable the learned representations to preserve both microscopic and macroscopic information.

To the best of our knowledge, there are a large number of works which focus on node classification or link prediction. However, the objectives of these works are completely different from that of our work, which aims to learn vector representations of better quality for the nodes, while the node classification or link prediction tasks are only utilized to evaluate the quality of the learned vectors.

3 Problem definition

In this section, we formally define the studied problem. The input network is defined as $G = (V, E, T)$, where $V = \{v_i\}_{i=1,\dots,|V|}$ consists of a set of nodes, $e_{i,j} = (v_i, v_j) \in E$ is an edge encoding the linkage relationship between the nodes, and $t_{v_i} \in T$ is a text document associated with each node v_i. Here we formally define the problem of text-enhanced network representation learning:

Definition Given a network $G = (V, E, T)$, the problem of text-enhanced network representation aims to learn a low-dimensional vector $r_{v_i} \in R^k$ for each node v_i in the network, where k is expected to be much smaller than $|V|$. The objective of text-enhanced network representation is to make the learned representation vectors explicitly preserve both network topology and text information of the nodes, so that the nodes close to each other in network topology or with similar text contents are close in the representation space.

4 Text-enhanced network representation

In this section, we present the details of the proposed text-enhanced network representation learning model. Firstly, Section 4.1 gives the introductions of Topology-Derived model. Secondly, Section 4.2 introduces Text-Derived model. Thirdly, TENR model based on the joint optimization of the above two models is introduced in details in Section 4.3. Finally, Section 4.4 presents complexity analysis of TENR model.

4.1 Topology-Derived model

Continuous Bag-of-Words model [21] called CBOW for short is a widely-used distributed word representation method. Following the idea of CBOW, we propose a novel negative sampling based model called Topology-Derived model, based on which we construct a set of node sequences by random walks generated from the network. Each node sequence can be regarded as a sentence in neural language models and each node in the network can be regarded as a word in neural language models. This model is composed of input layer, projection layer and output layer, which can predict the center node v_i given its contextual nodes $context(v_i) = v_{i-s}, v_{i-s+1}, \dots, v_{i+s-1}, v_{i+s}$, where s is the window size. In the projection layer,

$$X_{v_i} = \sum_{-s \leqslant j \leqslant s, j \neq 0} \boldsymbol{v}_{v_{i+j}}, \tag{1}$$

where both i and j are integers, and $\boldsymbol{v}_{v_{i+j}} \in R^k$ is the representation vector corresponding to the node v_{i+j}, where k is the vector dimension. Figure 1 shows Topology-Derived model.

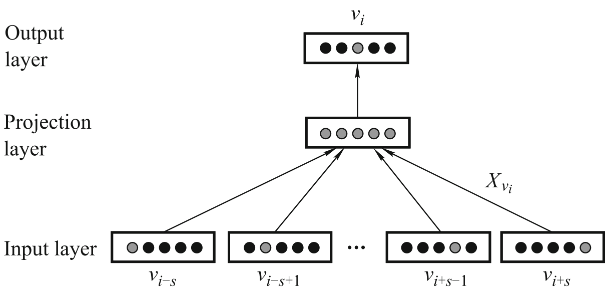

Fig. 1 Topology-Derived model

In this model, given a center node v_i and its contextual nodes $context(v_i)$, the node v_i is regarded as the positive sample and $NEG(v_i)$ is the set of negative samples of the center node v_i with a predefined size ds. For $\forall u \in V$, the labels of the node are defined as follows.

$$L^{v_i}(u) = \begin{cases} 1, & u \in \{v_i\}, \\ 0, & u \in NEG(v_i), \end{cases} \tag{2}$$

$p(u|context(v_i))$ defines the probability of predicting the node u given the contextual nodes $context(v_i)$. We try to solve the following probability.

$$maximize g_1(v_i) = \prod_{u \in \{\{v_i\} \cup NEG(v_i)\}} p(u|context(v_i)). \tag{3}$$

For each node v_i in V, we design two corresponding vectors: the embedding vector and the parameter vector. The embedding vector $\boldsymbol{v}_{v_i}$ is the representation of the node v_i when it is treated as the contextual node, while the parameter vector $\boldsymbol{\theta}_{v_i}$ is the representation of v_i when it is treated as the center node. $p(u|context(v_i))$ in Eq. (3) is defined as follows.

$$p(u|context(v_i)) = \begin{cases} \sigma(X_{v_i}^{\mathrm{T}} \boldsymbol{\theta}_u), & L^{v_i}(u) = 1, \\ 1 - \sigma(X_{v_i}^{\mathrm{T}} \boldsymbol{\theta}_u), & L^{v_i}(u) = 0, \end{cases} \tag{4}$$

where $\sigma(X_{v_i}^{\mathrm{T}} \boldsymbol{\theta}_u) = \frac{1}{1+e^{-X_{v_i}^{\mathrm{T}} \theta_u}}$ is a sigmoid function. X_{v_i} is the summing operation of the representation vectors corresponding to all nodes of $context(v_i)$. Eq. (4) can also be written as an integral expression.

$$p(u|context(v_i)) = [\sigma(X_{v_i}^{\mathrm{T}} \boldsymbol{\theta}_u)]^{L^{v_i}(u)} \cdot [1 - \sigma(X_{v_i}^{\mathrm{T}} \boldsymbol{\theta}_u)]^{1-L^{v_i}(u)}. \tag{5}$$

Consequently, Eq. (3) can be rewritten as follows.

$$\begin{aligned} maximize g_1(v_i) = \prod_{u \in \{\{v_i\} \cup NEG(v_i)\}} & [\sigma(X_{v_i}^{\mathrm{T}} \boldsymbol{\theta}_u)]^{L^{v_i}(u)} \cdot \\ & [1 - \sigma(X_{v_i}^{\mathrm{T}} \boldsymbol{\theta}_u)]^{1-L^{v_i}(u)}. \end{aligned} \tag{6}$$

Formally, maximizing $g_1(v_i)$ corresponds to maximizing the prediction likelihood of positive samples and minimizing the prediction likelihood of negative samples simultaneously, by which the network topology information is encoded into the node representation vectors.

4.2 Text-Derived model

The above Topology-Derived model is only based on the network topology to learn representations so that it cannot learn representations of nodes very well, which include rich text features, which may also be important to NRL. For example, the title of the paper regarded as a node of the citation network includes multiple words, which are regarded as text nodes associated with the paper. In order to learn better node representations, we incorporate the text nodes into the original network to construct a heterogeneous network. Figure 2 shows the heterogeneous network.

As shown in Fig. 2, the heterogeneous network is composed of two parts: the original network and text network. The original network consists of all circular nodes as well as the edges between these nodes. The text network consists of each circular node and the rectangular nodes associated with it as well as the edges between each circular node and its related rectangular nodes. Note that there are no edges between the rectangular nodes.

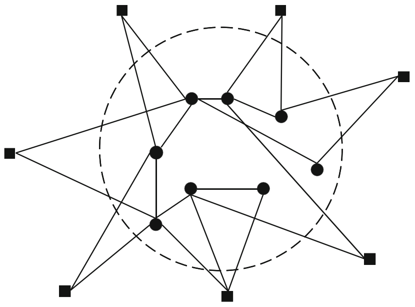

Fig. 2 Heterogeneous network

Let t_{v_i} denote the text node sets associated with the node $v_i \in V$. Our goal is to capture node-text semantic correlations by maximizing the probability of predicting the node given its related text nodes.

Inspired by the above Topology-Derived model, we regard the node v_i as a positive sample and other nodes not associated with the node v_i as the negative samples to construct a text-derived model. Suppose that the negative samples are defined as $NEG(v)$. For $\forall v \in t_{v_i}$, we define the labels of the nodes as follows.

$$\delta(\vartheta|v) = \begin{cases} 1, & \vartheta \in \{v_i\}, \\ 0, & \vartheta \in NEG(v), \end{cases} \tag{7}$$

where the labels of the positive sample and negative samples are equal to 1 and 0 respectively. As for the given samples, we aim at maximizing the following probability.

$$
\begin{aligned}
g_2(v_i) &= \prod_{v\in t_{v_i}} p(v_i|v) \\
&= \prod_{v\in t_{v_i}} \prod_{\vartheta\in\{\{v_i\}\cup NEG(v)\}} \left\{\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)^{\delta(\vartheta|v)} \cdot [1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]^{1-\delta(\vartheta|v)}\right\} \\
&= \prod_{v\in t_{v_i}} \left\{\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_{v_i}) \cdot \prod_{\vartheta\in NEG(v)} [1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\right\}, \qquad (8)
\end{aligned}
$$

where $\boldsymbol{e}_v$ is the parameter vector corresponding to the text node $v \in t_{v_i}$.

Formally, by maximizing $g_2(v_i)$, the node text features are encoded into the node representation vectors.

4.3 TENR model

As the first attempt, TADW incorporates the text features of nodes into network embedding process under the framework of matrix factorization. However, there are two limitations of TADW. Firstly, the very time and memory consuming of matrix factorization process of TADW makes it not scalable to large-scale networks. Secondly, TADW simply ignores the contexts of text features, so cannot appropriately capture semantic correlations between the nodes and their related words.

In order to address the above issues, we propose a novel model named as Text-Enhanced Network Representation Learning called TENR for short, which is regarded as Topology-Derived model plus Text-Derived model. Compared to TADW, TENR is improved at two levels: (1) at the network topology level, TENR exploits inter-node relationships by maximizing the probability of predicting the node given its contextual nodes in random walks generated from the network; (2) at the node text level, TENR captures node-text semantic correlations by maximizing the probability of predicting the node given its related words. By means of the above improvements, TENR model is expected to address the above issues to get the vector representations of better quality. Figure 3 shows TENR framework.

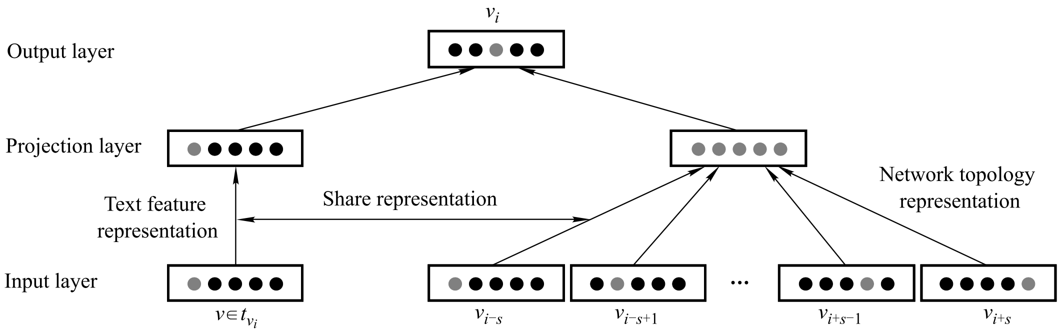

Fig. 3 TENR framework

As shown in Fig. 3, the network topology representation and text feature representation learned by Topology-Derived model and Text-Derived model respectively share the same representation, which can comprehensively utilize the contexts and text features of each node to get the vectors of better quality.

On the basis of TENR model, we construct a corpus C, which is a set of node sequences by random walks generated from the network. For ease of calculation, take the logarithm of $g_1(v_i)$ and $g_2(v_i)$, and based on the corpus C, we aim at maximizing the following joint objective probability function of TENR model.

$$
\begin{aligned}
L &= \sum_{v_i\in C} \left\{\begin{array}{l} \sum_{u\in\{\{v_i\}\cup NEG(v_i)\}} \left\{L^{v_i}(u)\cdot\log[\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)] + [1-L^{v_i}(u)]\cdot\log[1-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\right\}+ \\ \beta\cdot\sum_{v\in t_{v_i}}\sum_{\vartheta\in\{\{v_i\}\cup NEG(v)\}} \left\{\delta(\ \vartheta|\ v)\cdot\log[\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)] + [1-\delta(\ \vartheta|\ v)]\cdot\log[1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\right\} \end{array}\right\} \\
&= \sum_{v_i\in C} \left\{\begin{array}{l} \sum_{u\in\{\{v_i\}\cup NEG(v_i)\}} \left\{L^{v_i}(u)\cdot\log[\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)] + [1-L^{v_i}(u)]\cdot\log[1-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\right\}+ \\ \sum_{v\in t_{v_i}}\sum_{\vartheta\in\{\{v_i\}\cup NEG(v)\}} \beta\cdot\left\{\delta(\ \vartheta|\ v)\cdot\log[\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)] + [1-\delta(\ \vartheta|\ v)]\cdot\log[1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\right\} \end{array}\right\}, \qquad (9)
\end{aligned}
$$

where β is a harmonic factor to balance Topology-Derived model and Text-Derived model.

For ease of derivation, we define $L(v_i, u, v, \vartheta)$ as follows.

$$
\begin{aligned}
L(v_i, u, v, \vartheta) = \{&L^{v_i}(u)\cdot\log[\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)] + [1-L^{v_i}(u)]\cdot \\
&\log[1-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\} \\
&+\beta\cdot\{\delta(\ \vartheta|\ v)\cdot\log[\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)] \\
&+[1-\delta(\ \vartheta|\ v)]\cdot\log[1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\}. \qquad (10)
\end{aligned}
$$

And then we utilize stochastic gradient ascent to optimize the joint objective function L. The key is to give four kinds of gradients of L.

Firstly, we calculate the gradient on $\boldsymbol{\theta}_u$ of $L(v_i, u, v, \vartheta)$.

$$\begin{aligned}\frac{\partial L(v_i,u,v,\vartheta)}{\partial \boldsymbol{\theta}_u} &= L^{v_i}(u)\cdot[1-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\cdot X_{v_i}\\ &\quad -[1-L^{v_i}(u)]\cdot\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)\cdot X_{v_i}\\ &= \{L^{v_i}(u)\cdot[1-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\\ &\quad -[1-L^{v_i}(u)]\cdot\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)\}\cdot X_{v_i}\\ &= [L^{v_i}(u)-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\cdot X_{v_i}. \end{aligned} \tag{11}$$

Consequently, the updating formula of $\boldsymbol{\theta}_u$ is denoted as follows.

$$\boldsymbol{\theta}_u = \boldsymbol{\theta}_u + \alpha\cdot[L^{v_i}(u)-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\cdot X_{v_i}, \tag{12}$$

where α is the learning rate of TENR model.

Secondly, we calculate the gradient on X_{v_i} of $L(v_i, u, v, \vartheta)$. We use the symmetry property between $\boldsymbol{\theta}_u$ and X_{v_i} to get the gradient on X_{v_i}.

$$\frac{\partial L(v_i,u,v,\vartheta)}{\partial X_{v_i}} = [L^{v_i}(u)-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\cdot\boldsymbol{\theta}_u. \tag{13}$$

Consequently, the updating formula of $\boldsymbol{v}_{v'}$ is denoted as follows, where $v' \in context(v_i)$.

$$\begin{aligned}\boldsymbol{v}_{v'} &= \boldsymbol{v}_{v'} + \alpha\cdot\sum_{u\in\{\{v_i\}\cup NEG(v_i)\}}\frac{\partial L(v_i,u,v,\vartheta)}{\partial X_{v_i}}\\ &= \boldsymbol{v}_{v'} + \alpha\cdot\sum_{u\in\{\{v_i\}\cup NEG(v_i)\}}[L^{v_i}(u)-\sigma(X_{v_i}^{\mathrm{T}}\boldsymbol{\theta}_u)]\cdot\boldsymbol{\theta}_u. \end{aligned} \tag{14}$$

Thirdly, we calculate the gradient on $\boldsymbol{\theta}_\vartheta$ of $L(v_i, u, v, \vartheta)$.

$$\begin{aligned}\frac{\partial L(v_i,u,v,\vartheta)}{\partial \boldsymbol{\theta}_\vartheta} &= \beta\cdot\Big\{\frac{\partial}{\partial\boldsymbol{\theta}_\vartheta}\{\delta(\vartheta|v)\cdot\log[\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\\ &\quad +[1-\delta(\vartheta|v)]\cdot\log[1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\}\Big\}\\ &= \beta\cdot\{\delta(\vartheta|v)\cdot[1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\cdot\boldsymbol{e}_v\\ &\quad -[1-\delta(\vartheta|v)]\cdot\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)\cdot\boldsymbol{e}_v\}\\ &= \beta\cdot\{\{\delta(\vartheta|v)\cdot[1-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\\ &\quad -[1-\delta(\vartheta|v)]\cdot\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)\}\cdot\boldsymbol{e}_v\}\\ &= \beta\cdot[\delta(\vartheta|v)-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\cdot\boldsymbol{e}_v. \end{aligned} \tag{15}$$

Consequently, the updating formula of $\boldsymbol{\theta}_\vartheta$ is denoted as follows.

$$\boldsymbol{\theta}_\vartheta = \boldsymbol{\theta}_\vartheta + \alpha\cdot\beta\cdot[\delta(\vartheta|v)-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\cdot\boldsymbol{e}_v. \tag{16}$$

Finally, we calculate the gradient on $\boldsymbol{e}_v$ of $L(v_i, u, v, \vartheta)$. We use the symmetry property between $\boldsymbol{\theta}_\vartheta$ and $\boldsymbol{e}_v$ to get the gradient on $\boldsymbol{e}_v$.

$$\frac{\partial L(v_i,u,v,\vartheta)}{\partial \boldsymbol{e}_v} = \beta\cdot[\delta(\vartheta|v)-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\cdot\boldsymbol{\theta}_\vartheta. \tag{17}$$

Consequently, the updating formula of $\boldsymbol{e}_v$ is denoted as follows, where $v \in t_{v_i}$, $1 \leqslant i \leqslant |V|$.

$$\boldsymbol{e}_v = \boldsymbol{e}_v + \alpha\cdot\beta\cdot[\delta(\vartheta|v)-\sigma(\boldsymbol{e}_v^{\mathrm{T}}\boldsymbol{\theta}_\vartheta)]\cdot\boldsymbol{\theta}_\vartheta. \tag{18}$$

We utilize stochastic gradient ascent (SGA) method for optimization. In our implementation, we approximate the effect of β through instance sampling (node-node and node-text) in each training epoch. More details are shown in Algorithm 1.

Algorithm 1 TENR

```
Input:
    Network $G = (V, E, T)$
    Embedding size $d$
Output:
    Embedding matrix $X \in R^{|V|\times d}$
for node $v_i$ in $V$ do
    initialize embedding vector $\boldsymbol{v}_{v_i} \in R^{1\times d}$
    initialize parameter vector $\boldsymbol{\theta}_{v_i} \in R^{1\times d}$
    for node $v$ in $t_{v_i}$ do
        initialize parameter vector $\boldsymbol{e}_v \in R^{1\times d}$
    end for
end for
node sequences $C$ = RandomWalk()
for $(v_i, context(v_i))$ in $C$ do
    update parameter vectors following Formula (12)
    update embedding vectors following Formula (14)
    update parameter vectors following Formula (16)
    for node $v$ in $t_{v_i}$ do
        update parameter vectors following Formula (18)
    end for
end for
for $i = 0; i < |V|; i++$ do
    $X_i = \boldsymbol{v}_{v_i}$
end for
return $X$
```

Algorithm 1 adopts the same random walk as DeepWalk, where the number of walks to start at each node is 10, and the length of walks to start at each node is 40.

4.4 Complexity analysis

As a popular Topology-Derived method, DeepWalk utilizes the hierarchical softmax method to reduce the computational complexity of calculating the probability of a pair of node-context in the random walk node sequence from $O(|V|)$ to $O(\log|V|)$. Consequently, the computational complexity of DeepWalk is $O(|C|\cdot 2w\cdot\log|V|)$, where w is the window size of the contextual nodes. In the objective function (9), the computational complexity is further reduced to $O(|C|\cdot(ds+1)\cdot(\beta\cdot M+1))$, where the computational complexities of Topology-Derived and Text-Derived model are $O(|C|\cdot(ds+1))$ and

$O(|C| \cdot (ds+1) \cdot M)$, where ds is the predefined size of the negative sample sets and also a constant number irrelevant to the size of the network, and $M = max\{|t_{v_1}|, |t_{v_2}|, \ldots, |t_{v_{|V|}}|\}$, which is the maximum number of the text node sets t_{v_i}. Compared to DeepWalk, TENR model is faster.

5 Experiments

5.1 Datasets

We select three real-world network datasets as follows:

Citeseer is a subset of CiteSeerX data, which is a typical paper citation network. This dataset consists of 4610 scientific publications from 10 distinct research areas and 5923 edges, which are citation relationships between them.

DBLP is also a paper citation network, which consists of bibliographic data in computer science. This dataset consists of 17725 conference papers from four research areas and 105781 edges between them in total.

Weibo is a broadcast-style social network platform in China, which shares the brief and real-time information through the concern mechanism. This dataset consists of 62095 active users from 12 different concerned topics and 1383025 edges, which are friendships between them.

The detailed statistics are listed in Table 1.

Table 1 Statistics of datasets

Datasets	Citeseer	DBLP	Weibo
Nodes	4610	17725	62095
Edges	5923	105781	1383025
Labels	10	4	12
Average degree	2.57	11.94	44.55

5.2 Baseline methods

DeepWalk. DeepWalk is a popular network topology-only representation learning method, which uses local information obtained from truncated random walks to learn vector representations by treating walks as the equivalent of sentences.

node2vec. node2vec is an algorithmic framework for learning vector representations for nodes in networks, whose innovation is to improve the strategy of random walk to explore neighborhood architecture.

LINE. LINE is proposed to learn network representations for large scale networks, which takes into account both 1-order and 2-order proximity, and the concatenation of these two representations is used as the final embedding.

HARP. HARP is a method for learning low-dimensional vector representations to preserve higher-order structural features by compressing the input graph prior to embedding it. HARP in this experiment is the meta-strategy algorithm to improve DeepWalk.

Text. We take text matrix $T \in R^{|V| \times 100}$ as 100-dimensional network representation. The method is content-only baseline.

DeepWalk + Text. We simply concatenate the vectors from DeepWalk and text features into a 200-dimensional vector for network representations.

TADW. TADW is proposed to learn low-dimensional vector representations by incorporating text features of nodes into network representation under the framework of matrix factorization.

CANE. CANE is proposed to learn context-aware network representations for nodes with mutual attention mechanism and model semantic relationships between the nodes.

TENR@1. TENR is proposed to learn network representations, which preserves the features of the text nodes containing all stop words.

TENR@2. TENR is proposed to learn network representations, which preserves the features of the text nodes deleting all stop words.

5.3 Classifiers and experiment setup

We conduct the experiments on three real-world datasets. We adopt node classification tasks to verify the feasibility of our method. For all three datasets, we reduce the dimension of vectors to 100. To evaluate our method, we randomly select a portion of datasets as training set, and the rest is testing set. We take representation vectors of nodes as input to train classifiers, and calculate the accuracies of node classifications based on different training ratios, which range from 10% to 90%.

5.4 Experimental results and analysis

The node classification results for three datasets are shown in Tables 2–4. TENR consistently outperforms other baseline methods on different datasets, which shows the feasibility of our method.

The Micro-F1 and Macro-F1 values for multi-label classification on three datasets are reported in Tables 2–4. From the three tables, we have the following interesting observations:

(1) TENR consistently outperforms all of the other baseline methods on all three datasets. For example, TENR achieves the best performance and beats the best baseline CANE on Citeseer and DBLP, while it outperforms the best baseline DeepWalk+Text on Weibo. In addition, TENR outperforms the remaining baselines more or less to some extent.

Table 2 Node classification results on Citeseer

Training ratio		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1 (%)	DeepWalk	56.80	59.53	60.64	61.26	60.93	62.47	63.59	63.02	63.56
	node2vec	61.05	63.79	64.21	65.06	65.81	66.70	67.21	65.62	64.75
	LINE	29.93	33.46	34.41	35.03	34.79	34.90	36.15	38.45	35.57
	HARP	61.17	64.20	65.25	65.98	66.56	67.50	68.54	66.78	66.10
	Text	64.77	67.25	68.83	70.23	71.48	71.72	72.45	72.34	73.10
	DeepWalk+Text	63.81	68.34	70.68	71.57	72.32	73.24	73.04	73.67	73.51
	TADW	76.12	77.45	78.40	78.79	79.02	79.15	79.28	79.00	78.86
	CANE	76.75	77.83	78.37	78.85	78.89	79.20	79.21	79.07	78.98
	TENR@1	78.50	79.88	80.94	81.38	81.82	82.42	82.79	82.65	83.30
	TENR@2	79.01	81.48	81.84	82.65	82.86	83.08	83.80	83.19	83.73
Macro-F1 (%)	DeepWalk	34.18	35.90	36.83	37.03	36.81	37.75	38.44	38.31	38.55
	node2vec	36.81	38.58	38.81	39.33	39.86	40.40	40.72	39.90	39.34
	LINE	16.54	19.24	20.31	20.76	20.73	20.80	21.97	23.51	22.49
	HARP	36.92	39.85	38.90	40.82	40.78	41.12	41.85	40.80	40.15
	Text	39.58	40.98	41.98	42.82	43.58	43.75	44.20	44.06	44.64
	DeepWalk+Text	40.11	41.50	42.89	43.36	43.85	44.31	44.49	44.23	45.04
	TADW	46.78	47.69	48.70	48.57	49.10	49.23	48.69	48.77	48.56
	CANE	46.57	47.76	48.66	48.82	49.07	49.52	48.78	48.38	48.73
	TENR@1	47.82	48.73	49.84	50.93	51.81	52.46	52.04	51.94	52.65
	TENR@2	48.17	49.67	51.01	52.26	52.57	53.50	53.75	53.16	54.02

Table 3 Node classification results on DBLP

Training ratio		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1 (%)	DeepWalk	75.89	76.32	76.93	77.01	77.08	77.04	77.39	77.18	76.54
	node2vec	77.92	78.48	78.96	79.11	79.05	79.03	79.31	79.69	79.44
	LINE	54.29	55.97	56.79	56.84	56.71	57.24	57.60	58.01	57.05
	HARP	78.48	79.35	79.57	79.92	80.16	80.04	79.81	79.63	79.41
	Text	67.99	68.80	69.45	69.75	69.78	70.25	70.48	71.50	72.02
	DeepWalk+Text	77.61	78.25	79.39	79.45	79.53	79.66	79.78	79.27	79.60
	TADW	80.05	80.51	80.64	80.87	81.09	81.38	81.31	81.13	80.82
	CANE	80.24	81.08	81.29	81.34	81.55	81.66	81.68	81.31	81.75
	TENR@1	80.09	81.20	81.45	81.56	81.67	81.70	81.73	81.09	81.80
	TENR@2	80.62	81.36	81.50	81.70	81.76	81.85	82.78	82.34	82.36
Macro-F1 (%)	DeepWalk	69.27	69.61	70.42	70.81	70.79	70.68	71.05	70.64	70.31
	node2vec	70.93	72.23	72.82	73.06	73.13	73.22	73.69	74.17	73.72
	LINE	37.10	41.11	42.84	43.15	43.41	44.13	44.74	45.26	44.77
	HARP	70.86	72.56	73.19	73.81	74.25	74.28	73.99	73.67	73.76
	Text	59.00	60.51	61.64	62.03	62.17	62.52	62.90	63.09	63.74
	DeepWalk+Text	70.95	72.76	73.29	73.74	74.10	74.44	74.50	74.80	74.94
	TADW	72.59	73.91	74.81	74.96	75.10	75.38	75.12	75.32	75.43
	CANE	73.35	74.59	75.06	75.02	75.28	75.47	75.33	75.45	75.60
	TENR@1	73.43	74.65	74.98	75.81	75.90	75.40	75.30	75.90	76.04
	TENR@2	73.80	75.28	75.73	75.96	75.98	76.38	76.10	76.19	76.36

These experimental results demonstrate that TENR is effective and robust.

(2) From the three tables, we find that a simple concatenation of representation vectors from DeepWalk and Text, has better performances than Deepwalk or Text on the three datasets, which shows the importance of both the network topology structure and text features.

(3) The best baseline CANE and the second best baseline TADW almost have the same performance on the small-scale citation network Citeseer and DBLP. However, TADW has better performance than CANE, and weaker performance than DeepWalk+Text on the large-scale social network Weibo, which verifies that TADW is not scalable to large-scale networks and CANE is not general since it

Table 4   Node classification results on Weibo

Training ratio		10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1 (%)	DeepWalk	40.19	40.69	40.89	41.09	41.13	41.15	41.20	40.68	40.81
	node2vec	40.22	40.75	41.10	41.24	41.50	41.84	41.57	41.42	41.58
	LINE	39.48	39.98	40.24	40.40	40.59	40.62	40.78	40.50	40.68
	HARP	40.76	41.15	41.95	42.00	42.23	42.35	42.16	42.25	41.89
	Text	71.16	72.20	72.43	72.81	72.77	72.80	73.03	72.91	72.99
	DeepWalk+Text	73.20	74.39	74.56	75.02	75.29	75.14	75.40	75.45	75.52
	TADW	64.94	67.90	69.40	70.47	71.07	71.59	72.04	72.45	72.68
	CANE	40.70	41.26	42.05	42.56	42.45	42.78	43.02	42.80	42.47
	TENR@1	73.74	74.99	75.26	75.54	75.68	75.90	76.21	76.39	76.30
	TENR@2	74.29	75.19	75.61	75.82	76.03	76.30	76.52	76.68	76.57
Macro-F1 (%)	DeepWalk	34.28	34.78	35.01	35.20	35.18	35.29	35.33	34.99	35.14
	node2vec	34.41	34.90	35.26	35.45	35.60	35.74	35.48	35.29	35.39
	LINE	33.00	33.77	33.95	34.13	34.42	34.39	34.49	34.45	34.66
	HARP	35.09	35.61	36.04	36.14	36.42	36.59	36.41	35.92	35.89
	Text	70.94	71.16	71.44	71.80	71.72	71.90	72.08	71.96	71.69
	DeepWalk+Text	72.05	73.37	74.18	74.50	74.62	74.33	74.51	74.88	75.16
	TADW	63.08	66.65	68.37	69.46	70.15	70.72	71.28	71.69	71.84
	CANE	35.12	35.60	35.98	36.25	36.45	36.74	36.58	36.02	36.46
	TENR@1	72.50	74.23	74.59	75.06	75.11	74.54	74.70	75.00	75.31
	TENR@2	73.06	74.58	74.85	75.17	75.40	75.45	75.42	75.53	75.84

cannot be used when there is rich text information on Weibo.

(4) TENR@2, which deletes the stop words from the text nodes associated with each node of the original network, has a little better classification performance than TENR@1, which contains the stop words, which demonstrates that these stop words interfere with the vector representations.

From these observations we find that TENR generates high-quality representations by considering the network topology and text features simultaneously. Moreover, TENR is not task-specific and the vector representations can be conveniently used for different tasks, such as link prediction, similarity computation.

5.5 Parameter sensitivity

TENR model has a hyper-parameter: harmonic factor β to balance Topology-Derived model and Text-Derived model. We fix the training ratio to 50% and test Micro-F1 and Macro-F1 values of TENR@1 and TENR@2 with different β.

We let β vary from 0.1 to 0.9 on Citeseer, DBLP and Weibo datasets. Figure 4 shows the comparisons of Micro-F1 and Macro-F1 values of TENR@1 and TENR@2 with different β. Figs. 4(a), (b) and (c) are the comparisons of Micro-F1 values, and Figs. 4(d), (e) and (f) are the comparisons of Macro-F1 values. From Fig. 4, we find that as the value of the parameter β increases, the changes of Micro-F1 and Macro-F1 values follow different trends on all three datasets, but all the change ranges are within 1%, which shows that overall the performance of TENR model is not very sensitive to the parameter β, demonstrating the robustness of our model. Overall, for the Citeseer dataset, the best evaluated results in terms of Micro-F1 and Macro-F1 are achieved at $\beta = 0.3$. For the DBLP dataset, the best evaluated results in terms of Micro-F1 and Macro-F1 of TENR@1 are achieved at $\beta = 0.5$, and the best evaluated results in terms of Micro-F1 and Macro-F1 of TENR@2 are achieved at $\beta = 0.7$. For the Weibo dataset, the best evaluated results in terms of Micro-F1 and Macro-F1 of TENR@1 are achieved at $\beta = 0.7$, and the best evaluated results in terms of Micro-F1 and Macro-F1 of TENR@2 are achieved at $\beta = 0.3$.

5.6 Visualizations

We propose TENR to learn network representations on Citeseer, DBLP and Weibo. To demonstrate whether the representation vectors generated from TENR show discriminative clustering abilities or not, we randomly select four network categories on Citeseer and Weibo, each of which includes 150 nodes and three network categories on DBLP, each of which includes 200 nodes. Figure 5 shows node clustering visualizations on Citeseer, DBLP and Weibo.

As shown in Fig. 5, Figs. 5(a), (b) and (c) are the clustering visualization results of TENR@1. Figs. 5(d), (e) and (f) are the clustering visualization results of TENR@2. From Fig. 5, we can find that TENR learns efficient representation vectors with better

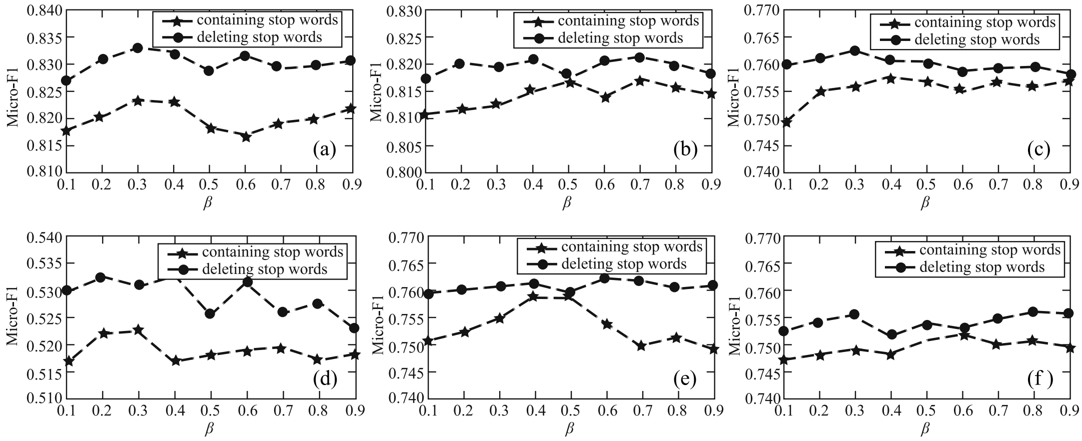

Fig. 4 Parameter sensitivity. (a) Citeseer; (b) DBLP; (c) Weibo; (d) Citeseer; (e) DBLP; (f) Weibo

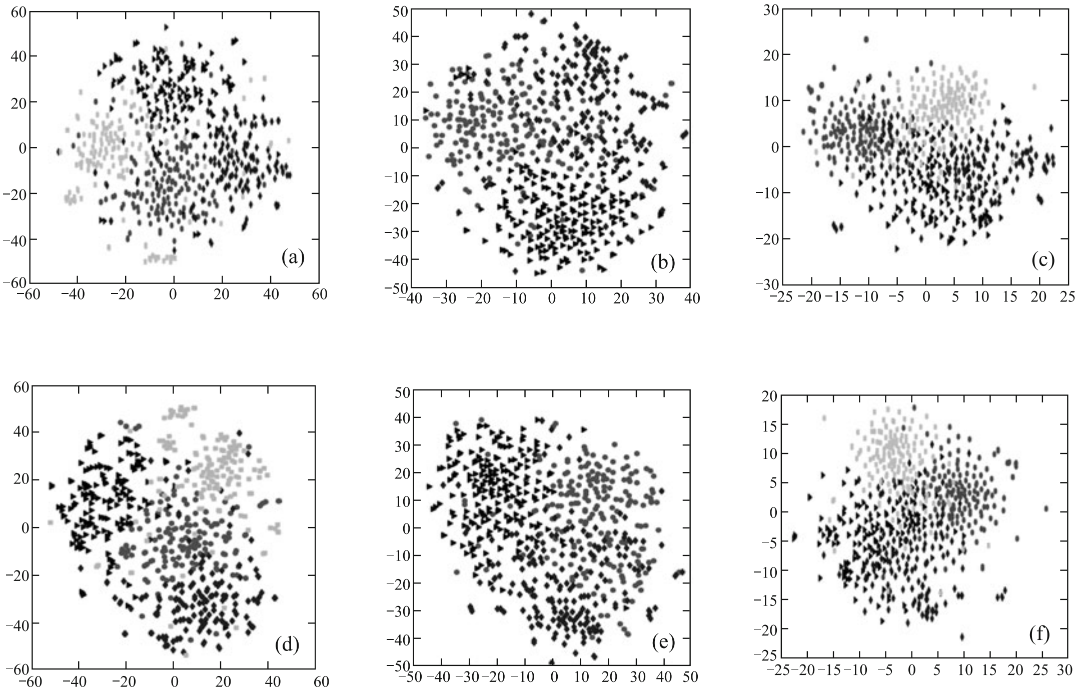

Fig. 5 Clustering visualizations. (a) Citeseer; (b) DBLP; (c) Weibo; (d) Citeseer; (e) DBLP; (f) Weibo

clustering abilities. The representation vectors from TENR@2 on Citeseer, DBLP and Weibo datasets show stronger clustering abilities than from TENR@1, and the boundaries between the categories on Figs. 5(d), (e) and (f) are clearer and more discriminative than on Figs. 5(a), (b) and (c), because the stop words disturb the vectors. In addition, the representation on Weibo dataset shows a relatively weaker clustering ability than Citeseer and DBLP datasets. This reason is that there are more network links in Weibo network, which leads to closer spatial distances among the vectors obtained from TENR than Citeseer and DBLP networks. In a word, the results of visualization demonstrate the effectiveness of our model.

5.7 Case study

To verify the performance of TENR, we conduct an experiment on Citeseer dataset. The selected document title is “exploiting population information in evolutionary learning”. As shown in Table 5, by using the representation vectors generated from DeepWalk, TADW, TENR@1 and TENR@2, we find three nearest documents to the selected document ranked by cosine similarity. We find that all these documents are cited by the selected document or some of these documents cite the selected document. Three nearest documents by TENR@1 and TENR@2, whose similarities to the selected document are higher than DeepWalk and TADW,

Table 5 Three nearest documents found by DeepWalk, TADW and TENR

Method	Title	Similarity
DeepWalk	an evolutionary approach to the automatic design of ensembles of neural network classifiers	0.8522
	hybrid soft computing systems a critical survey with engineering applications	0.8318
	evolutionary artificial neural networks	0.8276
TADW	how to make best use of evolutionary learning	0.7753
	making use of population information in evolutionary artificial neural networks	0.7595
	evolutionary ensembles with negative correlation learning	0.7116
TENR@1	making use of population information in evolutionary artificial neural networks	0.8768
	meta-learning evolutionary artificial neural networks	0.8440
	trends in evolutionary robotics	0.8409
TENR@2	how to make best use of evolutionary learning	0.9126
	making use of population information in evolutionary artificial neural networks	0.9018
	evolutionary artificial neural networks	0.8772

totally contain a relevant word to the selected document, such as "evolutionary". This indicates that TENR can learn better network representations with the help of text features than DeepWalk and TADW.

6 Conclusion

In this paper, we propose Text-Enhanced Network Representation Learning, which is a novel and discriminative network representation method to take the network topology and text features together into consideration. We conduct experiments with the task of node classification on three datasets (Citeseer, DBLP and Weibo). The experimental results show that TENR is an effective and robust network representation method compared to other baseline methods. Meanwhile, the visualization results of network representations generated by TENR demonstrate strong discrimination ability. TENR provides a normalized framework for joint learning with different types of resources. For future work, we will explore some new methods to incorporate community features of nodes into network representation learning.

Acknowledgements The work was supported by the National Natural Science Foundation of China (Grant Nos. 11661069 and 61763041), the Program for Changjiang Scholars and Innovative Research Team in Universities (IRT_15R40).

References

1. Tsoumakas G, Katakis I. Multi-label classification: an overview. International Journal of Data Warehousing and Mining, 2007, 3(3): 1–13
2. Tu C C, Liu Z Y, Sun M S. Inferring correspondences from multiple sources for microblog user tags. In: Proceedings of the 3rd Chinese National Conference on Social Media Processing. 2014, 1–12
3. Yu H F, Jain P, Kar P, Dhillon I S. Large-scale multi-label learning with missing labels. In: Proceedings of the 31st International Conference on Machine Learning. 2014, 593–601
4. Libennowell D, Kleinberg J M. The link-prediction problem for social networks. Journal of the Association for Information Science and Technology, 2007, 58(7): 1019–1031
5. Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014, 701–710
6. Grover A, Leskovec J. Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, 855–864
7. Tang J, Qu M, Wang M Z, Zhang M, Yan J, Mei Q Z. Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. 2015, 1067–1077
8. Cao S S, Lu W, Xu Q K. Grarep: learning graph representations with global structural information. In: Proceedings of the 24th International Conference on Information and Knowledge Management. 2015, 891–900
9. Wang D X, Cui P, Zhu W W. Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016, 1225–1234
10. Chen H C, Perozzi B, Hu Y F, Skiena S. HARP: hierarchical representation learning for networks. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence. 2018, 2127–2134
11. Yang C, Liu Z Y, Zhao D L, Sun M S, Chang E Y. Network representation learning with rich text information. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence. 2015, 2111–2117
12. Tu C C, Liu H, Liu Z Y, Sun M S. CANE: context-aware network embedding for relation modeling. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017, 1722–1731
13. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems. 2013, 3111–3119
14. Tu C C, Zhang W C, Liu Z Y, Sun M S. Max-Margin DeepWalk: discriminative learning of network representation. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. 2016,

3889–3895

15. Tu C C, Zeng X K, Wang H, Zhang Z Y, Liu Z Y, Sun M S, Zhang B, Lin L Y. A unified framework for community detection and network representation learning. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(6): 1051–1065
16. Li C Z, Wang S Z, Yang D J, Li Z J, Yang Y, Zhang X M, Zhou J S. PPNE: property preserving network embedding. In: Proceedings of the 22nd International Conference on Database Systems for Advanced Applications. 2017, 163–179
17. Yang D J, Wang S Z, Li C Z, Zhang X M, Li Z J. From properties to links: deep network embedding on incomplete graphs. In: Proceedings of the 26th ACM International Conference on Information and Knowledge Management. 2017, 367–376
18. Sun X F, Guo J, Ding X, Liu T. A general framework for content-enhanced network representation learning. 2016, arXiv preprint arXiv:1610.02906
19. Pan S R, Wu J, Zhu X Q, Zhang C Q, Wang Y. Tri-party deep network representation. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. 2016, 1895–1901
20. Zhou H, Zhao Z Y, Li C, Liang Y Q, Zeng Q T. Rank2vec: learning node embeddings with local structure and global ranking. Expert Systems with Applications, 2019, 136: 276–287
21. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: Proceedings of the 1st International Conference on Learning Representations. 2013

Yu Zhu received his MS from Chang'an University, China in 2012. He is a PhD student in Qinghai Normal University as well as a teacher in Qinghai University at present. His research interests include data mining and machine learning.

Zhonglin Ye received his MS from Southwest Jiaotong University, China in 2016. He received his Doctor of Engineering Degree from Shaanxi Normal University, China in 2019. He is a teacher in Qinghai Normal University at present. His research interests include data mining and machine learning.

Haixing Zhao received his Doctor of Engineering Degree from School of Computer Science in Northwestern Polytechnical University, China in 2004, and also received his Doctor of Science Degree from University of Twente, Holland in 2005. He is a professor and part-time professor in Qinghai Normal University and Shaanxi Normal University respectively. His research interests include complex networks and applications, machine translation and machine learning.

Ke Zhang received his MS from Qinghai Normal University, China in 2014, and received his Doctor of Engineering Degree from Qinghai Normal University, China in 2019. He is a teacher in Huzhou University at present. His research interests include hyper-graph theory and complex network.