2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)

Semi-supervised Deep Network Representation with Text Information

Xinchun Ming
School of Information Science and Technology
University of Science and Technology of China
Hefei 230027,China
Email: yzu2ustc@gmail.com

Fangyu Hu
School of Information Science and Technology
University of Science and Technology of China
Hefei 230027,China
Email: hufy@ustc.edu.cn

***Abstract*—Network representation learning aims at learning low-dimensional representation for each vertex in a network, which plays an important role in network analysis. Conventional shallow models often achieve sub-optimal network representation results for non-linear network characteristics. Most network representation methods merely concentrate on structure but ignore text information related to each node. In the paper, we propose a novel semi-supervised deep model for network representation learning. We adopt a random surfing model to capture the global structure and incorporate text features of vertices based on the PV-DBOW model. The joint similarity between vertices achieved by combining network structure and text information is applied as the unsupervised component. While the first-order proximity in a network is used as the supervised component. By jointly optimizing them, our method can obtain reliable low-dimensional vector representations. The experiments on two real-world networks show that our method outperforms other baselines in the task of multi-class classification of vertices.**

Keywords-network representation learning; semi-supervised model; deep model; text features

I. Introduction

Relations between numerous entities in real life can be abstracted as various complex networks, e.g., citations between academic papers, friendship between online social networks or co-purchase relations between commodities. Rich information such as structural relationship or vertex attribute exists in these networks. Mining the information within these networks plays an important role in many network analysis applications, including community discovery[16], vertex classification[12] and link prediction[13]. However, these tasks face a series of problems such as data sparsity, high non-linearity and informational multiplicity. To address these challenges, network representation learning (NRL) encodes each vertex in a network into a low-dimensional space, which is an effective way in mining information in networks, then machine learning algorithms can be applied directly.

Recently, majority network representation methods have been proposed. Perozzi[10] proposed a DeepWalk model which utilizes the second-order proximity and transforms networks into collections of linear sequences by truncated random walks. Tang[14] developed a novel large-scale information network embedding method, which preserves both the first-order and second-order proximities. Luo[8] stated that linear manifolds cannot be applied to the network representations accurately as the underlying structure of real data is often highly nonlinear. To solve this problem, Wang[18] proposed a Structural Deep Network Embedding model (SDNE) with multiple layers of non-linear functions. The method outperforms other conventional shallow models such as IsoMAP[15], Laplacian Eigenmaps[11] and LINE[14] in graph embedding. The above methods merely take the local network structure as input but ignore text information related to each node.

In real world, many network applications often have rich text information, such as the abstract or title information of each paper in a citation network and substantial text information in web pages. Content information associated with each node is also crucial in network representation learning. A method named Text-associated DeepWalk (TADW)[19] incorporates text features of vertices into network representation learning under the framework of matrix factorization. But TFIDF matrix does not consider the contextual information of a document, which always results in suboptimal representation.

In this paper, we propose a Semi-supervised Deep Network Representation with Text Information model, namely semi-TDNR, which takes both structural information and content information associated to each node into consideration. We first revisit the structural deep network embedding model[18], which uses the second-order proximity as the unsupervised component and uses the first-order proximity as the supervised component. Inspired by this method, we consider using a random surfing model which displaces the second-order proximity to capture global structure information. In this way, we can directly yield a probabilistic co-occurrence matrix instead of adjacency matrix. To incorporate text features of vertices, we adopt a paragraph vector model[6] named PV-DBOW, which is an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts. Then structure-text matrix applied as the unsupervised component can be achieved by combining structural co-occurrence matrix and text cosine similarity matrix. Next, we apply a deep learning method that is similar to SDNE to the structure-text matrix and

978-1-5386-1829-5/17/$31.00

achieve low-dimensional representation for each vertex in a network. The experiments on two real-world networks show that the proposed model achieves superior performance than other baselines in the task of multi-class classification of vertices.

The contributions of our work can be summarized as follows:

- We propose a Semi-supervised Deep Network Representation with Text Information method, namely semi-TDNR, to perform network representation learning. The method is able to preserve the global network structure and incorporate text information related to each node.
- We develop a novel semi-supervised deep model, which simultaneously optimizes the first-order proximity and the structure-text similarity matrix. The representations for network vertices are robust to sparse networks.
- The proposed method is extensively evaluated on two real datasets in the task of multi-class classification of vertices. The results show that the model achieves superior performance than other baselines.

II. Background and Related Work

A. Deep Neural Network

Deep neural network is extensively used to learn multiple levels of feature representations in image classification[20], speech recognition[3] and natural language processing[5]. Restricted boltzmann machines[2] and deep autoencoder[4] provide a novel greedy layer-wise pre-training method to train these networks. However, little work based on deep neural network has been devoted to consider in NRL. DeepWalk[10] constructs a corpus generated by random walks from the network and achieves representation for each vertex based on the skip-gram model. Wang[18] proposed a semi-supervised deep model, which exploits the first-order and second-order proximities to effectively capture the highly non-linear structure. Motivated by this method, we apply the semi-supervised deep model to our structure-text matrix to learn low-dimensional representation. The framework of this work can be seen in Figure 2.

B. Random Walk

Random walk is an effective way to capture network structural information. The method first randomly selects one vertex from the network, then select the next vertex from all the neighbours of the current vertex with probabilities related to edge-weight. Random walks have achieved effective results in many domains such as link prediction, community detection and NRL. Liu[7] proposed a method based on local random walk which has a much lower computational complexity in the task of link prediction. Zhu[21] combined the data generation component of the random walk and the inferential component of the HDP topic model to detect the network communities. Perozzi[10] transformed unweighted network structural information into linear sequences by truncated random walk for learning network representation. In our work, we adopt a random surfing model and construct a structural co-occurrence matrix to capture global network structure.

C. Text Vector Representation

Nowadays, many methods exist to represent a sentence or document to a vector space. TFIDF describes the importance of a term by combining the term frequency in the given document and a document frequency of the term in the whole collection of documents. Blei[1] introduced a topic model called LDA to represent the probability distribution on all topics for each document in a corpus. However, these models don't consider the contextual information of a document. To solve this problem, Mikolov[9] developed a simple neural network without non-linear hidden layers to learn distributed vectors for words. In this paper, we adopt a distributed bag of words model proposed by Le[6] to represent each document associated with each node by a dense vector. Then, the text similarity between vertices can be achieved by calculating the cosine similarity between these text vectors.

III. Semi-TDNR Model

In this section, we first define the problem. Then we introduce the proposed Semi-TDNR model including random surfing, text features extraction and loss functions. At last, we describe the algorithm of learning representation for each vertex in a network.

A. Problem Definition

A information network is denoted as $G = (V, E, T)$, where V represent the members of the network and E is the edge relationship between the nodes. T represent the collections of the texts related to vertices. The network representation learning aims at learning low-dimensional representation $u_v \in R^k$ (k is a small number of latent dimensions) for each vertex $v \in V$. Vertices close to each other in network topology or with similar text information are close in the representation space.

B. The Model

1) Random Surfing: DeepWalk transforms networks into collections of linear sequences by truncated random walks and learns representation based on the skip-gram model. However, the method merely utilizes the second-order proximities, which can't capture global structural information. Motivated by the PageRank model used for ranking tasks in web pages, we adopt a random surfing model to capture the global structure of the network. Now, we have a transition matrix A that captures the transition probabilities between different vertices. The row vector p_t indicates the probability of reaching the j-th vertex after t steps of transitions from the i-th vertex. p_0 is an initial vector with the 1-hot information,

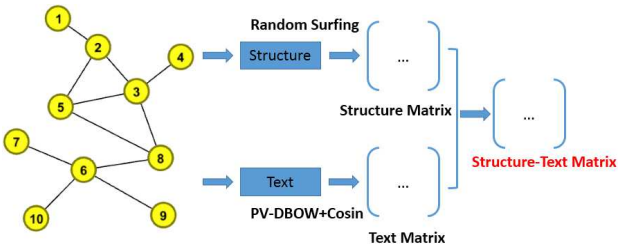

Figure 1. The process of achieving structure-text matrix

while the value of the *i*-th entry is 1 and other entries are 0. Then we have the following recurrence relation:

$$p_t = \alpha \cdot p_{t-1} A + (1-\alpha) p_0 \tag{1}$$

As we can see the structure matrix from Figure 1, a probabilistic co-occurrence matrix S consists of the row vector p_t associated with each vertex is yielded after t steps of transitions. The matrix indicates the structural similarity between different vertices.

2) Text Features Extraction: For the conventional methods of text vector representation such as TFIDF or LDA cannot capture the contextual information of text related to each vertex, we adopt the PV-DBOW model[6] to achieve text features. According this method, we have the following objective functions:

$$\mathcal{L}_t = \sum_{i=1}^{N} log P(w_{-b} : w_b | v_i) \tag{2}$$

$$P(w_j | v_i) = \frac{exp(v_{v_i}^{\mathrm{T}} v_{w_j}^{'})}{\sum_{w=1}^{W} exp(v_{v_i}^{\mathrm{T}} v_{w_j}^{'})} \tag{3}$$

where $w_{-b} : w_b$ is a context window of length b and v_i is a text paragraph id related to the vertex i. The context windows in the same paragraph have the common paragraph id. $v_{w_j}^{'}$ is the representation of word w_j. The output representation of text $v_{v_i}^{\mathrm{T}}$ associated with each node can be achieved by maximizing Eq. 3.

3) Loss Functions: In the above introduction, we have achieved the probabilistic co-occurrence matrix S and the text feature related to each node in the network. We can also obtain the text similarity matrix T by computing the cosine similarity between different text features. To merge network structure and text information in a network, as we see from Figure 1, the structure-text matrix M devoted to the input of our deep neural network is generated by the following equation:

$$M = \beta S + (1-\beta) T \tag{4}$$

where β indicates the proportion of the structural similarity and the text similarity. Then we apply a stacked denoising autoencoder (SDAE) model[17] on structure-text matrix M

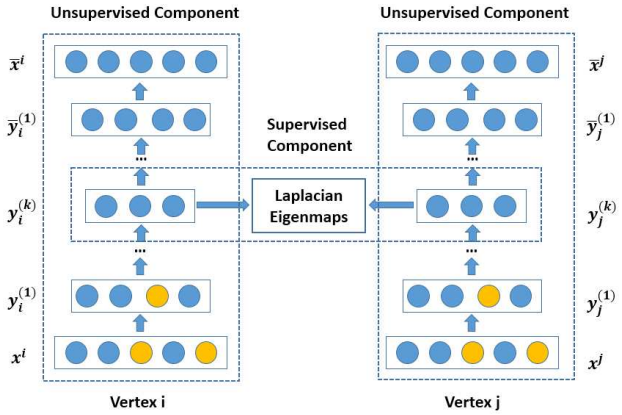

Figure 2. The framework of applying a semi-supervised SDAE model to structure-text matrix

regarded as the unsupervised component. As the yellow neurons shown in the Figure 2, the model partially corrupts the input data before taking the encoding step, which is different from conventional autoencoders and enhance the robustness of our model. Deep autoencoder model consists of two parts, i.e. the encoder and the decoder. The encoder transforms the vector in the input space to a new feature space with an activation function, while the decoder reconstructs the original input space back from the latent representation space. We first make some definitions on some parameters to be convenient to represent the loss function. X=$\{x_i\}_{i=1}^{n}$ is the input data and $Y^{(k)}$=$\{y_i^{(k)}\}_{i=1}^{n}$ is the k-th layer hidden representations. $\overline{X}$=$\{\overline{x}_i\}_{i=1}^{n}$ is the reconstructed data. The hidden representations for each layer are shown as follows:

$$\begin{aligned} y_i^{(1)} &= \delta(W^{(1)} x_i + b^{(1)}) \\ y_i^{(k)} &= \delta(W^{(k)} y_i^{k-1} + b^{(k)}), k = 2, ..., K \end{aligned} \tag{5}$$

The loss function of the unsupervised reconstruction error is shown as follows:

$$\mathcal{L}_u = \sum_{i=1}^{n} \| \overline{x}_i - x_i \|_2^2 \tag{6}$$

Motivated by the SDNE model, we consider the first-order proximity regarded as the supervised component to capture the local information and constrain the similarity of the latent representations between vertices belong to the same edge. The loss function is defined as follows:

$$\mathcal{L}_s = \sum_{i,j=1}^{n} s_{i,j} \| y_i - y_j \|_2^2 \tag{7}$$

where y_i or y_j is the output representations of the vertex i or j. $s_{i,j}$ is equal to 1 if the vertex i and j belong to the same edge in the network. Our semi-TDNR model combines the unsupervised component and the supervised component to capture both local and global information of the network.

Similar to the SDNE model, we can obtain the following loss function by combining Eq. 6 an Eq. 7:

$$\mathcal{L} = \mathcal{L}_u + \gamma \mathcal{L}_s \tag{8}$$

At last, the output representation for each vertex with structural and text information can be achieved by minimizing Eq. 8.

C. The Algorithm of Our Model

The algorithm of the semi-supervised deep network representation with text information model can be described as Alg. 1.

Algorithm 1 semi-TDNR(G, α, β, γ)

Input: the network G = (V, E, T)
restart random surfing parameter α
input matrix parameter β
loss function parameter γ
Output: Network representations Y
updated Parameters: W

1: initialized parameters $W = \{w^{(1)}, ..., w^{(K)}\}$
2: Based on random surfing and PV-DBOW, apply Eq. 4 to obtain input matrix M
3: **repeat**
4: Based on Eq. 8, use $\partial\mathcal{L}/\partial w$ to obtain updated parameters W by the back-propagate algorithm.
5: **until** convergence
6: Obtain the network representations $Y = Y^{(K)}$

IV. Experiments

In this section, we conduct the experiments on two real datasets to assess the performance of our semi-TDNR model. After achieving the low-dimensional representation for each vertex in a network, we evaluate the effectiveness of the network representations in the task of multi-class classification of vertices.

A. Datasets

- **Cora** contains 2211 machine learning papers from 7 classes and 5001 links between them. The links represent the citation relationships between papers. Besides structural information, we extract the abstract of each paper that is used as the text information associated with each vertex in the network.
- **Citeseer** consists of scientific publications from 10 distinct research areas including agriculture, archaeology, biology, computer science, financial economics, industrial engineering, material science, petroleum chemistry, physics, and social science. The network contains 10 classes with 3312 vertices and 4675 edges in total. The edges are citation relationships between the documents. Besides these, we regard the article title as the text information related to each node.

B. Baseline Algorithms

We use the following five methods as the baseline algorithms:

- DeepWalk[10]: It adopts truncated random walks to capture the structural information. The network representations are generated by skip-gram model.
- Doc2Vec[6]: It is a paragraph vector model which learns fixed-length feature representations for texts based on an unsupervised algorithm. Here we use PV-DBOW model to represent the text associated with each vertex in the network.
- SDNE[18]: It is a semi-supervised deep model which merely utilizes the network structure. The method preserves the first-order proximity used as the supervised component and the second-order proximity used as the unsupervised component.
- TADW[19]: It incorporates text features of vertices into network representation learning under the framework of matrix factorization. The text features are generated by applying SVD to TFIDF matrix.
- Semi-DNR: It only utilizes structural information comparing with semi-TDNR. The model uses the structural co-occurrence matrix achieved by random surfing as the input of the semi-supervised deep model.

C. Parameter Settings

We propose a semi-supervised deep model with multiple layers of non-linear functions in this paper. The dimension of each layer for different datasets is listed in Table 1. Besides this, we offer other parameters such as random surfing coefficient α, input matrix coefficient β and loss function coefficient γ in this table.

Table I
The parameters of our Semi-TDNR model

Datasets	Neural layers	α	β	γ
Cora	2211-500-100	0.98	0.3	0.5
Citeseer	3312-1000-500-100	0.98	0.7	0.5

D. Classifiers and Experiment Setup

In this section, we report the results of the network representations in the task of multi-class classification of vertices. We use linear SVM model as supervised classifier and take representations of vertices as features to train this classifier. Then we adopt *Micro-F*1 and *Macro-F*1 to evaluate the performance of our classifiers. We first define TP_i, FP_i, FN_i as the number of true positives, false positives and false negatives in the instances which are predicted as *i*. F_i is the F-measure value for category *i*. The evaluation metrics can be shown as follows:

$$\pi = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FP_i)} \quad \rho = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i + FN_i)}$$

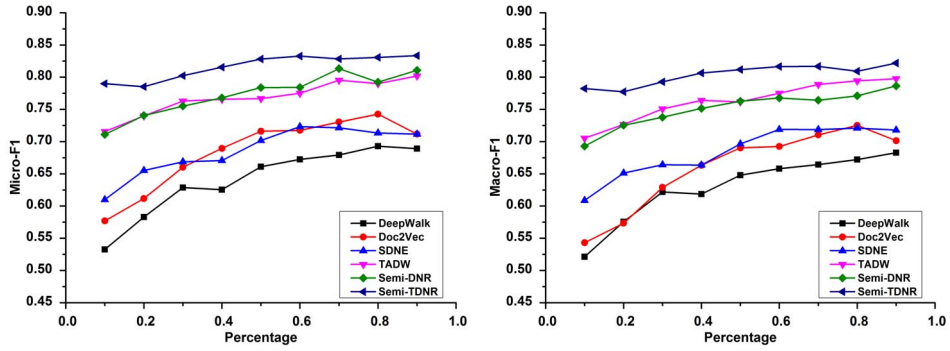

Figure 3. *Micro-F*1 and *Macro-F*1 on Cora

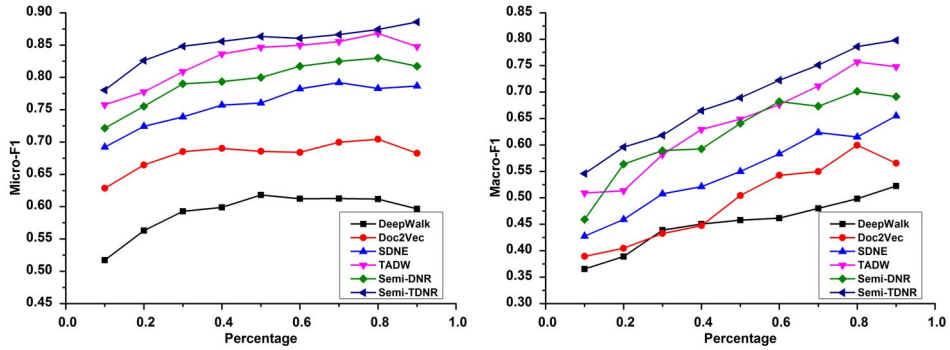

Figure 4. *Micro-F*1 and *Macro-F*1 on Citeseer

$$F_{micro} = \frac{2\pi\rho}{\pi + \rho} \tag{9}$$

$$F_{macro} = \frac{\sum_{i=1}^{M} F_i}{M} \tag{10}$$

We randomly select vertices in the citation networks as training set and the remaining vertices as test set. The training ratio varies from 10% to 90% for classifiers. Next, we repeat the step for 5 times and report the average *Micro-F*1 and *Macro-F*1. The classification results on Cora and Citeseer are shown in Figure 3 and Figure 4.

As these figures show, DeepWalk performs fairly poor on the citation networks. This is mainly because DeepWalk merely utilizes the second-order proximities which cannot capture sparse network structure. Doc2vec achieves much better results than DeepWalk for rich text information.

- **SDNE *vs*. Semi-DNR:** Semi-DNR is about average 10% on Cora and 5% on Citeseer more than SDNE. Although two methods concentrate on network structure, yet SDNE only consider local structure such as the first-order and the second-order proximities. Semi-DNR use global structure based on random surfing to generate representations.

- **Semi-DNR *vs*. Semi-TDNR:** Semi-TDNR is about average 6% on Cora and 8% on Citeseer more than Semi-DNR. It is mainly because Semi-TDNR takes network structure and content information related to each node into consideration. But Semi-DNR only utilizes the global structural information without other information.

- **Semi-TDNR *vs*. TADW:** Semi-TDNR gets about average 6% on Cora and 3% on Citeseer increase though two models incorporate both network structure and text information. On the one hand, TADW is a shallow model based on the framework of matrix factorization which cannot capture the non-linear network structure, while Semi-TDNR adopts a deep neural network model to capture this non-linear characteristics. On the other hand, TADW obtains text features based on the TFIDF matrix which cannot capture the contextual information. To solve this problem, Semi-TDNR adopts a paragraph vector model called PV-DBOW.

V. Conclusion

In this paper, we propose a semi-supervised deep network representation with text information model. The model adopts a stack denoising autoencoder model to capture the highly non-linear network structure. To address the informational multiplicity challenge, we incorporate text features of vertices based on PV-DBOW into learning network repre-

sentation. The structure-text matrix achieved by combining structural co-occurrence matrix and text cosine similarity matrix indicates the joint similarity between vertices. By jointly optimizes the structure-text matrix used as the unsupervised component and the first-order proximity used as the supervised component, our method can preserve the global network structure and incorporate text information associated with each vertex. The experiments on two real-world networks show that our method outperforms other baselines in the task of multi-class classification of vertices.

For future work, our direction is to explore the network representation learning of large-scale network data. Besides this, we will try to adopt other deep learning technologies such as convolutional neural network or recurrent neural network to our work.

REFERENCES

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] Kostadin Georgiev and Preslav Nakov. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *ICML (3)*, pages 1148–1156, 2013.

[4] Mojtaba Gholamipour and Babak Nasersharif. Mapping mel sub-band energies using deep belief network for robust speech recognition. In *Telecommunications (IST), 2016 8th International Symposium on*, pages 510–514. IEEE, 2016.

[5] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[6] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[7] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[8] Weiping Liu and Linyuan Lü. Link prediction based on local random walk. *EPL (Europhysics Letters)*, 89(5):58007, 2010.

[9] Dijun Luo, Feiping Nie, Heng Huang, and Chris H Ding. Cauchy graph embedding. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 553–560, 2011.

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

[11] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. *Network*, 11(9):12, 2016.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[13] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[14] Xiaoxiao Shi, Yao Li, and Philip Yu. Collective prediction with latent graphs. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1127–1136. ACM, 2011.

[15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. ACM, 2015.

[16] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[17] Cunchao Tu, Hao Wang, Xiangkai Zeng, Zhiyuan Liu, and Maosong Sun. Community-enhanced network representation learning for network analysis. *arXiv preprint arXiv:1611.06645*, 2016.

[18] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[19] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1225–1234. ACM, 2016.

[20] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. Network representation learning with rich text information. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[21] Matthew D Zeiler. *Hierarchical convolutional deep learning in computer vision*. PhD thesis, New York University, 2013.