Semi-Supervised Heterogeneous Information Network Embedding for Node Classification using 1D-CNN

Nasrullah Sheikh
Dept. of Information Engineering and Computer Science
University of Trento
Sommarive Povo, Trento, Italy
nasrullah.sheikh@unitn.it

Zekarias T. Kefato
Dept. of Information Engineering and Computer Science
University of Trento
Sommarive Povo, Trento, Italy
zekarias.kefato@unitn.it

Alberto Montresor
Dept. of Information Engineering and Computer Science
University of Trento
Sommarive Povo, Trento, Italy
alberto.montresor@unitn.it

***Abstract*—Network Representation Learning (NRL) is a method to learn a representation of a graph in a low-dimensional space, such that the representation can be later utilized easily in various machine learning tasks such as classification, recommendation, and prediction. In contrast to homogeneous networks, *heterogeneous information networks* (HINs) contain rich semantics and structural information due to multiple types of nodes and edges. Due to heterogeneity, the conventional representation learning methods are not directly applicable. In this paper, we propose a semi-supervised HIN embedding model, adopted from the natural language processing community. The model uses sequences of nodes obtained by random walks constrained on edge types such that the structural and semantic properties are preserved. These sequences correspond to sentences in a document. Each sequence is labeled based on the nodes contained in it. We adopt a 1D-Convolutional Neural Network sentence classification model that seeks to fit a sequence classifier while optimizing the representation of the nodes. We have performed experiments on vertex classification on two widely used real-world datasets, showing better or comparable performance with respect to the state-of-the-art.**

***Index Terms*—Heterogeneous Information Networks, Network Embedding, Deep Learning, Semi-Supervised Learning**

I. INTRODUCTION

Graphs capture the interactions between nodes in domains as diverse as social networks, protein networks and linguistics. Various machine learning tasks can be applied to them, including node classification [20], link prediction [16], recommendation [7], and visualization [24]. The performance of these tasks depend on the input features of the graphs. Recently, various network representation learning (NRL) approaches have been proposed to capture the information present in the network and encode it in a low-dimensional vector space, while preserving the relationships in the learned space.

Several of these NRL approaches, such as DEEPWALK [19] and NODE2VEC [3], have taken inspiration from language models [15], [17]. These approaches treat sequences of nodes as analogous to sentences in a document and then apply language models such as SKIPGRAM [17] to learn an embedding of nodes.

Existing NRL approaches have focused on embedding *homogeneous* networks, where only one type of nodes and edges exist. Real-world networks are by and large inherently heterogeneous - having distinct types of nodes and edges.

For example, consider a patent dataset as shown in Fig. 1, which consists of three different nodes: patent, inventor and assignee. A patent has semantically different relationships in the network. An inventor invents a patent, the patent is assigned to an assignee, and a patent cites other patents. Links connect both different and similar types of nodes. Due to different semantics of node relationships, the homogeneous representation learning methods cannot be applied because they cannot preserve the different semantics.

In this paper, we propose HETNET2VEC, a novel approach to learn a representation of nodes in a heterogeneous information network (HIN). To preserve the semantic relationships, we perform relationship-specific short random walks in the network. As in various homogeneous representation learning approaches, random walks [9], [19], [25] are used to preserve the contextual relationships between nodes. Random walk sequences are analogous to sentences. We train a sequence classifier based on a 1D-Convolutional Neural Network (CNN). The goal is to learn *node embeddings* that enable accurate sequence classification, while at the same time preserve the semantic relations encoded in the multi-modal walks. Various works show that CNNs perform quite well in various NLP tasks such as sentiment classification [13]

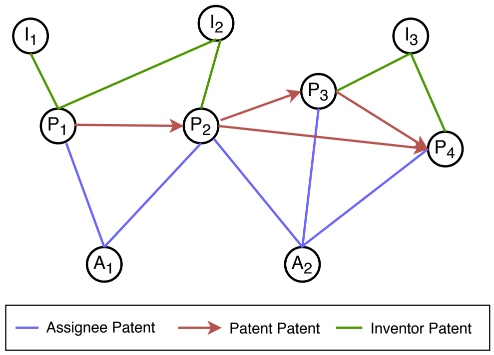

Fig. 1. An Example of Patent HIN

and answer selection [30]. We adopt the model from the work of Kim [13] on sentence classification due to the stark resemblance of the inputs and optimization objective. The model is a $1D$-CNN with multiple filters taking embedded vectors of random walk sequences as input.

The rest of the paper is organized as follows: Section II discusses the related work. Section III briefly describes HINs, and formally defines the problem. In Section IV, we describe the data preparation step (IV-A) and the design of the model (IV-B). Experiments are described in Section V. Finally, we conclude in Section VI.

II. Related Work

The aim of network representation learning is to embed nodes in a low-dimensional space. Traditional approaches [1], [27], [28] focus on the factorization of the network matrix, which can be obtained through various methods such as adjacency matrix, Laplacian matrix, or Katz similarity. Graph factorization [1] methods factorize an adjacency matrix of the graph to obtain the node embeddings. Tang et al. [28] uses normalized Laplacian matrix and selects the k-smallest eigenvectors to represent node embeddings. These methods suffer from high computational costs.

Recent advances in representation learning in various domains such as computer vision and NLP have been ported to the graph domain [9], [18], [19], [21], [22], [26], [29]. In their seminal paper, Perozzi et al. proposed DeepWalk [19] which uses SkipGram [17], a neural network word embedding model, to learn representation of nodes in a network. They used short random walks to generate sequences of vertices which are analogous to the sentences in a document and employ SkipGram to learn on these node sequences. Proposed by Grover et al., Node2Vec uses parameterized random walks to capture the neighborhood [9]. Line separately captures first order (1-hop) and second order (2-hop) proximities to learn two embedding of nodes [26]. These models use only structural information of graphs. Various studies [11], [18], [19], [22] have shown that using both the attribute and the content information helps in learning more precise embeddings. Apart from utilizing network specific information, Kefato et al. [14] proposed to incorporate diffusion event information in learning an embedding.

All these models have been proposed for homogeneous networks. Alternative approaches [4], [5], [8], [12], [25] have been proposed for representation learning on HINs. Based on meta-path based proximity, Huang et. al proposed hine [12], that aims at minimizing the distributions of meta-path proximities and embedding space proximities. Metapath2Vec++, proposed by Yuxio et al. [5], uses meta-paths based random walks to generate heterogeneous neighborhoods, and defines a heterogeneous negative sampling method which normalizes the softmax function with respect to the node type. A recent approach Hin2Vec learns node and relationship representations using a neural network model [8]. The method uses meta-path-based random walks and a negative sampling approach. The neural network uses a logistic binary classifier to predict multiple relationship between two nodes and simultaneously learn a embedding of nodes and relationships.

III. Problem definition

A *heterogeneous information network* (HIN) is a graph $G = (V, E, f, g)$, where V is the set of nodes, $E \subseteq V \times V$ is the set of edges, $f : V \rightarrow T_V$ is a function mapping each node $v \in V$ to one node type in T_V, and $g : E \rightarrow T_E$ is a function mapping each edge $e \in E$ to one edge type in T_E.

Given a heterogeneous information network G, we aim to learn a function $\Phi : V \rightarrow \mathbb{R}^d$ which maps each node $v \in V$ to a low d-dimensional vector space $\mathbb{R}^d$, where $d << |V|$, such that the different structural relationships between nodes are preserved.

IV. Model

In this section, we discuss the methods for generating the sequences of nodes (Section IV-A) followed by the description of our CNN model (Section IV-B).

A. Sequence Generation and Labeling

Algorithms based on language modeling require a corpus of words and sentences. Therefore, in order to use such models on networks, we need to build a corpus of sequences of vertices from the graph. Approaches based on random walks have been widely used in homogeneous networks to generate such corpus [19]. In case of heterogeneous networks, Sun et al. have shown that random walks ignoring node and edge types are highly biased towards the high out-degree node types [23]. Meta-path based random walks have been proposed [5], [8] for generating the walk sequence to capture the heterogeneous node contexts. These random walks are guided by a meta-path scheme. Meta-path based random walks require a large number of walks per node and longer walks in order to capture the contexts of nodes.

Considering these issues, we propose a scheme based on edge-constrained random walks. Given an edge type $t_i \in T_E$, a random walker traverses the paths based on the edge type t_i and ignores other edge types to generate a sequence of nodes up to a length of l. Each sequence of nodes captures the semantic and structural relationships between the nodes with respect to t_i in HIN. We repeat this for each edge type in HIN. These random walks forms a corpus sequence S. The corpus S captures the overall structural and semantic relationships present in the HIN. For example, in the U.S. PATENTS heterogeneous network, we use PP (patent-patent), PA (patent-assignee), and PI (patent-inventor) edge types to generate the corpus. The edges which connect two different types of nodes form a bipartite structure. Following Fig. 1, $[A_1, P_2, A_2, P_2, A_2, \cdots]$, $[I_1, P_1, I_2, P_2, I_2, \cdots]$, and $[P_1, P_2, P_3, P_4]$ are examples of sequences generated from relation specific random walks using PA, PI, and PP edge types respectively. Random walks on bipartite graphs to obtain structural contexts has also been used in [22], [25].

Labels of sequences are assigned based on labels of nodes, given that sequences form a coherent structural and semantic context. We randomly select a fraction of nodes from V, and use their labels to form a label of a sequence. Let $W \subseteq V$ be a randomly selected fraction of nodes from V, and $\lambda : V \to \mathbb{L}$ is a label assignment function that assigns a label to a node v from a set of labels $\mathbb{L}$. The label for a sequence $s \in S$ is:

$$L_s = [l : \lambda(v), \forall v \in s \ and \ v \in W]$$

B. HETNET2VEC *Model*

We adopt the CNN model proposed by Kim [13] for representation learning of nodes. The architecture of the model is shown in Fig. 2. The model was proposed for sentence classification. The model has multiple channels, and each channel applies same set of multiple filters of different sizes. The model in Fig. 2 shows two filters of size 3 and 2. The selection of this model is motivated by the stark resemblance of our data input *vs* the model input and the optimization objective function. Instead of the words in sentence classification, we have nodes, and a sequence of nodes represents a sentence. The sequences of nodes are obtained by edge-guided random walks as described in Section IV-A. The model classifies the sequences while optimizing the representation of the nodes.

The model input is a sequence S_i of vertices of length n, padded if necessary, from the corpus S. Each vertex $v \in S_i$ is represented by an embedding vector $e_j = \Phi(v) \in \mathbb{R}^d$. A sequence is represented as a matrix M of embedding vectors of its constituent vertices as $M = [e_1, e_2, \cdots, e_n]$. Let $M[i : j]$ represent the concatenation of row vectors from i^{th} row to j^{th} row. In the convolutional layer, we apply multiple filters of different sizes, $w_k \in \mathbb{R}^{hd}$ where h is the height of the k^{th} filter, and d is the embedding dimension to produce a new feature. The feature $c_k[i]$ is obtained by applying the k^{th} filter on the sub-matrices of M as:

$$c_k[i] = f(w_k \cdot M[i : i + h - 1] + b)$$

where $i \in [1, \ldots, n - h + 1]$, f is non-linear activation function, such as *relu*, $\cdot$ is the dot product between the filter and concatenated vectors of matrix $M[i : i + h - 1]$, and b is the bias. The convolutional operator is applied multiple times to obtain a sequence of feature maps $c_k \in \mathbb{R}^{n-h+1}$. In case of a multi-channel architecture as shown in Fig. 2, each filter is applied to both channels and the respective feature maps are added.

Multiple feature maps of varying sizes are produced from different filters. The dimensionality of each feature map varies according to the filter size. We apply *max-pooling* [2] as downsampling strategy to obtain fixed-length vectors from feature maps. These vectors are concatenated to form a top-level feature vector which is followed by a fully connected softmax layer that produces a probability distribution over labels. At the dense layer, we apply dropout regularization as proposed by Hinton et al. [10]. Since, each sequence has a label which is multi-label. Therefore, the training objective of the model is to minimize the binary cross-entropy while optimizing the weight vectors. In case of the two-channel architecture, both channels have node vectors but one is static throughout the training and other is tuned via back-propagation. The parameters include embedding weights, layer weight, and bias term b are trained using the back-propagation method.

V. EXPERIMENTS

In this section, we present an evaluation of our proposed approach. We first provide the description of two real-world datasets and the state-of-the-art baseline methods. Then, we provide an experimental setup for evaluation through multi-class classification of nodes.

A. Datasets

We use two heterogeneous datasets for evaluation: U.S. Patents[1] and Yelp[2]. Table I summarizes some statistics of these datasets. Edges between nodes can be either undirected $(-)$ or directed $(\to)$.

a) U.S. PATENTS*:* is a 3-year (1998-2000) drug[3] related patent dataset obtained from United States Patent and Trademark Office. The network contains three different types of nodes, patent (P), Inventor (I) and Assignee (A). The network has patent citations $P \to P$, patent-inventor $P - I$, and patent-assignee $P - A$ as edges.

b) YELP*:* is a dataset on reviews of restaurants by customers. We extracted data of one year (2010) of restaurants serving at least one out of 10 cuisines[4]. The dataset has three types of nodes: restaurants (R), users (U), and city (C). There are three types of relationships between nodes: $U - U$ (user friendships), $R - U$ (user's reviews) and $R - C$ (restaurant's city).

B. Baselines

We evaluate our approach against state-of-the-art NRL methods. Since DEEPWALK and NODE2VEC methods are designed for a homogeneous network, we will treat all nodes

[1]http://www.patentsview.org/download/

[2]https://www.yelp.com/dataset/challenge

[3]Drug patent classes: 128, 351, 424, 433, 435, 514, 600, 601, 602, 604, 606, 607, 800

[4]American, Mexican, Italian, Chinese, Japanese, Thai, Indian, Canadian, Spanish, Greek

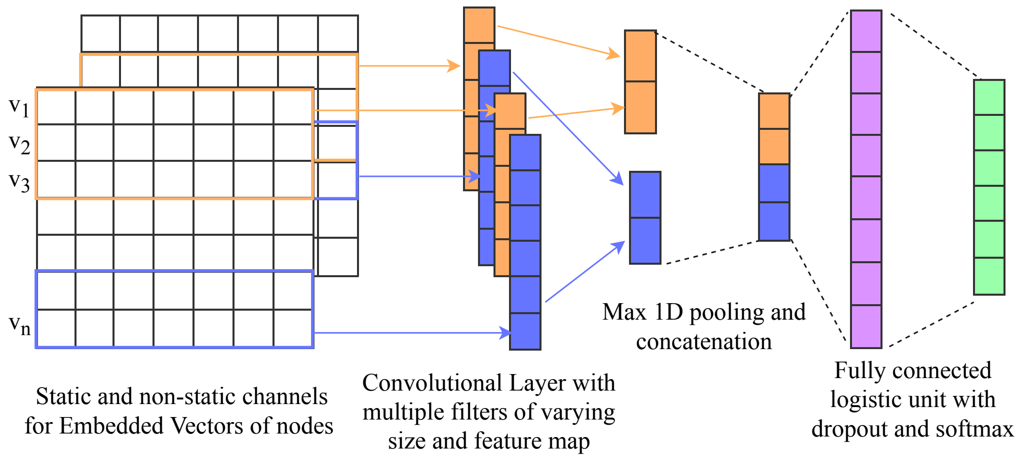

Fig. 2. The adopted 1D-CNN Architecture for Representation Learning in HIN

TABLE I
DATASET STATISTICS

Dataset	Nodes			Edges			Labels
USPATENTS	Patent	Inventor	Assignee	$P \rightarrow P$	$P - I$	$P - A$	14
	63,486	83,893	10,245	51,512	172,950	54861	
Yelp	User	Restaurant	City	$U - U$	$R - U$	$R - C$	10
	22,073	5914	10	424,344	56,699	5,914	

and edges in heterogeneous network as homogeneous ones to learn an embedding in both these approaches. The results from the baselines are obtained using the code released by the authors.

- DEEPWALK [19] is a random-walk based method to learn a d-dimensional node vectors.
- NODE2VEC [9] is a parameterized random walk based approach to generate the vertex sequences. The node embeddings are learned by using negative sampling in SKIPGRAM.
- HIN2VEC [8] is a single-hidden-layer neural network model to learn an embedding of nodes using meta-path based relationships between different nodes, and also learns an embedding for meta-paths in the heterogenous networks.

C. Experimental Setup

For the corpus generation we performed 10 random walks on each node in each relation, with a length of 80. For labelling the sequences, we used the labels of 50% nodes. Then we randomly selected a portion of these sequences for training. In case of a two-channel architecture, we initialized the non-static layer node vectors with the pre-trained WORD2VEC vectors, and the static layer is initialized randomly. In a single-channel architecture, the node vectors are randomly initialized. We employed 10 filters for each filter of size (3,5,7). The dense layer has *relu* activation, and the dropout rate is 0.2. The optimization is done using *Adam* with a learning rate of 0.001. The hyper-parameters for the model such as number and size of filters, number of epochs were selected by grid search.

The dimensionality of node vectors is set to 128 in all methods. The hyper-parameters for the baselines are selected as reported in the respective papers to get better performance.

D. Classification

After learning the node representation on the entire dataset, we perform patent and restaurant node classification in U.S. PATENTS and YELP datasets, respectively. In each dataset, we selected the nodes which were used in training the CNN model and use their representations as feature vectors for classification. These nodes comprise the training data and the remaining nodes are the test set for classifier. We trained a one-vs-rest logistic regression classifier, LIBLINEAR [6]. We report the average MICRO-F1 and MACRO-F1 scores as metrics for evaluation.

The results of node classification are shown in Table II. In U.S. PATENTS dataset, HETNET2VEC shows the improvement over the state-of-the-art models relatively by $8\% - 12\%$ and $4\% - 10\%$ in terms of MICRO-F1 and MACRO-F1 respectively. This indicates that the proposed model is able to preserve the different relationships in the HIN. Moreover, HETNET2VEC$_{rand}$, which is a single channel variant and the node vectors are initialized randomly, also performs better. In case of YELP dataset, the results are not too far from the baselines. This warrants further investigation in the model architecture and hyper-parameters.

TABLE II
MULTI-CLASS CLASSIFICATION ON PATENTS AND RESTAURANTS NODES

Metric	Method	MICROF1	MACROF1
U.S. PATENTS	DEEPWALK	0.47	0.39
	NODE2VEC	0.47	0.38
	HIN2VEC	0.49	0.41
	HETNET2VEC	**0.53**	**0.43**
	HETNET2VEC$_{rand}$	0.51	0.42
YELP	DEEPWALK	0.35	0.18
	NODE2VEC	0.36	0.17
	HIN2VEC	**0.36**	**0.19**
	HETNET2VEC	0.34	0.18

VI. CONCLUSION

In this work, we proposed a model based on CNN to learn a representation of nodes in a HIN. We captured the different relationships among nodes through edge-specific random walks. The model preserves the semantic and structural relationships between nodes. We showed empirically that the learned representations improve the node classification task. Future work in this direction include hyper-parameter optimizations, testing on various datasets, and applying other machine learning tasks such as link prediction.

REFERENCES

[1] A. Ahmed, N. Shervashidze, S. M. Narayanamurthy, V. Josifovski, and A. J. Smola. Distributed large-scale natural graph factorization. In *Proc. of the 22nd Int. World Wide Web Conf.*, WWW'13, pages 37–48, 2013.

[2] Y. L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2559–2566, June 2010.

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[4] S. Chang, W. Han, J. Tang, G.-J. Qi, C.C. Aggarwal, and T. S. Huang. Heterogeneous network embedding via deep architectures. In *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 119–128, 2015.

[5] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2Vec: Scalable representation learning for heterogeneous networks. In *Proc. of the 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '17, pages 135–144. ACM, 2017.

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[7] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):355–369, March 2007.

[8] T.-Y. Fu, W.-C. Lee, and Z. Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proc. of the 2017 ACM on Conf. on Information and Knowledge Management*, CIKM '17, pages 1797–1806. ACM, 2017.

[9] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proc. of the 22Nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864. ACM, 2016.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[11] X. Huang, J. Li, and Xia Hu. Label informed attributed network embedding. In *Proc. of the 10th ACM Int. Conf. on Web Search and Data Mining*, WSDM '17, pages 731–739. ACM, 2017.

[12] Z. Huang and N. Mamoulis. Heterogeneous information network embedding for meta path based proximity. *CoRR*, abs/1701.05291, 2017.

[13] Yoon K. Convolutional neural networks for sentence classification. In *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.

[14] Z.T. Kefato, N. Sheikh, and A. Montresor. Mineral: Multi-modal network representation learning. In *Machine Learning, Optimization, and Big Data - Third Int. Conf., MOD 2017, Volterra, Italy, September 14-17, 2017, Revised Selected Papers*, pages 286–298, 2017.

[15] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *CoRR*, 2014.

[16] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, May 2007.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[18] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang. Tri-party deep network representation. In *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence*, IJCAI'16, pages 1895–1901. AAAI Press, 2016.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proc. of the 20th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710. ACM, 2014.

[20] P. Sen, G.M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[21] B. Shaw and T. Jebara. Structure preserving embedding. In *Proc. of the 26th Annual Int. Conf. on Machine Learning*, ICML '09, pages 937–944. ACM, 2009.

[22] N. Sheikh, Z.T. Kefato, and A. Montresor. GAT2VEC: Representation learning for attributed graphs. *Computing*, Apr 2018.

[23] Y. Sun, J. Han, X. Yan, P.S. Yu, and T. Wu. PathSim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB*, 4(11):992–1003, 2011.

[24] J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proc. of the 25th Int. Conf. on World Wide Web*, WWW '16. Int. World Wide Web Conf.s Steering Committee, 2016.

[25] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proc. of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '15, pages 1165–1174. ACM, 2015.

[26] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. LINE: Large-scale information network embedding. In *Proc. of the 24th Int. Conf. on World Wide Web*, WWW '15, pages 1067–1077, 2015.

[27] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '09, pages 817–826. ACM, 2009.

[28] L. Tang and H. Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, Nov 2011.

[29] D. Wang, P. Cui, and W. Zhu. Structural deep network embedding. In *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, pages 1225–1234. ACM, 2016.

[30] T. Yang, W.-T. Yih, and C. Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2013–2018, 2015.