IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 6, 2018, accepted September 24, 2018, date of publication October 2, 2018, date of current version October 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2873044

Learning Heterogeneous Network Embedding From Text and Links

YUNFEI LONG[1], RONG XIANG[1], QIN LU[1], DAN XIONG[1], CHU-REN HUANG[2], CHENGLIN BI[3], AND MINGLEI LI[4]

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University, Hong Kong
[3]Advanced Micro Devices (Shanghai), Shanghai 201203, China
[4]Huawei Technologies Co., Ltd., Shenzhen 518100, China

Corresponding author: Minglei Li (minglei.li@connect.polyu.hk)

This work was supported by The Hong Kong Polytechnic University (PolyU RTVU) through GRF under Grant CERG PolyU 15211/14E and Grant PolyU 152006/16E.

ABSTRACT Finding methods to represent multiple types of nodes in heterogeneous networks is both challenging and rewarding, as there is much less work in this area compared with that of homogeneous networks. In this paper, we propose a novel approach to learn node embedding for heterogeneous networks through a joint learning framework of both network links and text associated with nodes. A novel attention mechanism is also used to make good use of text extended through links to obtain much larger network context. Link embedding is first learned through a random-walk-based method to process multiple types of links. Text embedding is separately learned at both sentence level and document level to capture salient semantic information more comprehensively. Then, both types of embeddings are jointly fed into a hierarchical neural network model to learn node representation through mutual enhancement. The attention mechanism follows linked edges to obtain context of adjacent nodes to extend context for node representation. The evaluation on a link prediction task in a heterogeneous network data set shows that our method outperforms the current state-of-the-art method by 2.5%–5.0% in AUC values with *p-value* less than 10^{-9}, indicating very significant improvement.

INDEX TERMS Network embedding, heterogeneous network, attention mechanism, text processing.

I. INTRODUCTION

Nowadays, networks are ubiquitous and many applications need to mine information within these networks. Network applications include DNA networks in biology [1]–[3], friendship/follower networks in social sciences [4]–[7], Internet of Things [8], [9], and word co-occurrence networks in linguistics [10]–[12], etc. With the wide use of networks in modeling and applications, network embedding, a method to use fixed dimension vectors to represent nodes in a network, becomes a hot research topic [6], [13]–[19].

According to the unitarity of node characters or features, there are two types of networks: homogeneous networks and heterogeneous networks. A homogeneous network only consists of one type of nodes and behaviors. A heterogeneous network contains different types of nodes. In many heterogeneous networks, especially user related networks, users (the subjects of behaviors) and products (the objects of behaviors) are commonly regarded as two different types of nodes.

Previous tasks in network embedding mostly focus on homogeneous networks. Different methods are proposed such as matrix factorization [20], random walk [14], and neural networks [21]. For homogeneous networks, the use of text information as context data is also common. Due to the complexity of this issues, few studies work on network embedding of heterogeneous networks. In heterogeneous networks, we consider it particularly important to leverage on both link information and other types of information such as text for product description and comments written by friends in social networks. Link information is considered naturally structured as a graph. Text information, on the other hand, is often free-structured or semi-structured [22].

To integrate link structure and text in the same network, two main issues need to be addressed: the first issue is how to learn node representation by integrating link information and text content coherently; the second issue is how to distinguish different types of nodes in the representation framework. This is particularly difficult if nodes are comprehensively related yet are of different types.

In this work, we propose a novel method to learn node representation in heterogeneous networks through the learnings of both link structure and available text content in a unified framework. To learn node representation, we sample each type of links separately to obtain conditional probabilities, and then the sampled edges are treated as binary links for model updating. This link embedding learning method is derived from the random walk method proposed by Tang *et al.* [23]. For text embedding, we propose to measure conditional probabilities of both link information and text information between any two nodes. We also propose a two-step neural network to process text not only at sentence level (individual user and product related comments), but also at document level (the collection of user and product related comments). In this way, we can obtain more comprehensive information including attentions and global semantics. A Recurrent Neural Network (RNN) model is used to assemble sentence level information, whereas an attention based Convolutional Neural Network (CNN) model is used to extract sentence-to-document level information. For the evaluation of heterogeneous networks, a large-scale heterogeneous network dataset is collected for embedding learning and will be made available for public access.

The contribution of this paper includes:

- A novel neural network based node representation model in a joint learning framework which incorporates both structured link information and unstructured text information in a hierarchical neural network. It has the capability to learn multiple types of nodes in a heterogeneous network.
- A novel hierarchical neural network model to obtain network embedding of text to include both sentence information and document information.
- A novel attention mechanism by extending the text of adjacent nodes through linked edges so that much larger context in the network can be included.
- Provision of an open accessed heterogeneous network dataset.

The evaluation on the link prediction in four benchmark datasets shows significant performance boost compared to state-of-the-art methods.

The rest of the paper is organized as follows: section II describes related works, including network embedding, neural network based text processing and attention models. Section III introduces our proposed method for joint learning of heterogeneous network embedding from link and text. Section IV elaborates the evaluation on various network embedding datasets to validate the effectiveness of our proposed method. Section V concludes this paper with future work direction.

II. RELATED WORK

Related work mainly include two parts. The first part is on network representation using embedding for both homogeneous and heterogeneous networks. The second part is on neural network models to process text information.

A. NETWORK EMBEDDING

Network embedding assigns nodes in a network to low-dimensional representations and effectively preserves the network structure. The represented network can either be homogeneous, which only contains a single type of nodes, or heterogeneous, which contains multiple types of nodes.

Network embedding traditionally uses matrix factorization. These algorithms represent connections between nodes in the form of a matrix and factorize the matrix to obtain the embedding according to the survey by Goyal and Ferrara [7]. Specific methods include Locally Linear Embedding (LLE) [24], Mixed Membership Stochastic Blockmodel (MSB) [25], and Graph Factorization [26]. However, matrix factorization is computationally expensive [27]. To address this issue, random walk methods are later used to approximate centrality [28] and similarity of network links [29]. Specific methods include Deepwalk [30], Line [23], and node2vec [14].

Random walk based methods are capable of processing large-scale networks. Studies show that fixed length random walks can preserve higher order proximity by maximizing sampled probabilities of occurrences of subsequent nodes [7]. Conventional random walks, however, mainly use first-order proximity. The first work to include the second order proximity features derived from conventional random walks was proposed by Tang *et al.* [23] for link prediction tasks.

The development of deep neural network has led to a deluge of deep neural network based methods for network representation tasks. Wang *et al.* [6] use a Structural Deep Network Embedding method (SDNE). As a semi-supervised deep model, SDNE has multiple layers of non-linear functions to capture non-linear features of a network by jointly optimizing first order proximity and second order proximity.

In many applications, a network not only contains links to other nodes, but also other information that indicates the properties of these nodes. Text information, such as the title of a product description or written comments made by users, is especially useful.

In contrast to homogeneous networks, a heterogeneous network has multiple types of nodes such as users and videos and multiple types of information associated with the nodes such as text, attributes, and multi-media contents. An effective approach is to embed nodes in a network into low dimensional vectors. When using node embedding as node representations, downstream tasks such as information retrieval, recommendation and node classification, etc., can be conducted in fixed dimensional space [6]. Long *et al.* [31] combine user and text information in the Hupu network for user preference identification. Complex methods, such as the Community-enhanced Network Representation (CENE) [32], leverage both network link information and text information by modeling text as a special kind of nodes, and then optimize the probabilities of heterogeneous links. Tu *et al.* [21] propose a state-of-the-art Context Aware Network Embedding (CANE) model to extract context information with an attention mechanism for text embedding. But CANE was proposed for a homogeneous network. For heterogeneous networks having multiple types of nodes, Gui *et al.* [33] used a large-scale network embedding model initially proposed by Tang *et al.* [23] to explore user and product representations. However, when text information is included, comments written by the same user at different times, or comments made by different users of the same product node are treated as isolated text units. Even though individual comments can be short, a collection of them, as a document set to each node, can give more comprehensive information of the node. There are yet methods to explore the use of document information in text embedding for the learning of network embedding. Chang *et al.* [34] demonstrate that the rich content and linkage information in a heterogeneous network can be captured by deep neural network approach, and the deep neural network is applied to represent heterogeneous network which contain both text and picture information.

B. NEURAL NETWORK BASED TEXT PROCESSING

In recent years, neural network based methods have noticeably improved the performance of Natural language processing (NLP) tasks. Commonly used models include CNNs [35], Recursive Neural Networks [36], and RNNs [37]. RNN naturally benefits sentiment classification because of its ability to capture sequential information in the text. However, standard RNN suffers from the gradient vanishing problem [38] where gradients may grow or decay exponentially over long sequences. To address this problem, Long-Short Term Memory model (LSTM) is introduced by adding a gated mechanism to keep long-term memory. Each LSTM layer is generally followed by mean pooling and then fed into the next layer. Experiments in datasets which contain both long sentences and long documents demonstrate that LSTM model outperforms the traditional RNN [39], [40].

In many NLP tasks, attention models are introduced to highlight the importance of certain semantic units such as key words and key sentences. Sentence level attention models, also referred to as local-context based attention models, aim to highlight the importance of informative words to a given sentence. Informative words are aggregated as attention weights to form sentence embedding representation. Aggregation of sentence level attentions, referred to as document level attentions, can give more comprehensive document embedding. Attention models are widely adopted in text classification tasks. Yang *et al.* [41] apply attention mechanism in document level sentiment classification. In other tasks, Yin *et al.* [42] propose an attention-based convolutional neural network for modeling sentence pairs. Tu *et al.* [21] propose the first model to introduce an attention mechanism to a network embedding task. In the work by Yang *et al.* [41], both sentence level attention and document level attention are proposed to further improve performance.

III. PROPOSED MODEL

Network nodes often have both link and text content regardless of its homogeneity. For easy explanation of the formalism used in this paper, we first introduce an animation video website Bilibili[1] as an example to demonstrate network heterogeneity and how text can be used in different perspectives. The Bilibili website has two types of nodes. User nodes, as one type of nodes, is generally involved with a friend/follower network. Text comments written by users are posted on a bulletin linking to specific animation videos. The collection of reviews by a particular user is a good source to find information about the user's personal preferences in addition to his/her subjective opinions of the videos. Video nodes, another node type related to animations, also have a collection of comments written by different users. The collection of reviews by different users for a given video reflect the collective opinions and should be more objective on the whole.

In general, a heterogeneous network G can be represented as a graph $G = (V, E, T)$, where V is the set of nodes, E is the set of edges, and T is the set of documents. Furthermore, V can be of different types. For easy illustration without loss of generality, let us assume a heterogeneous network has two types of nodes: user nodes and product nodes[2] denoted by u_i and a_k such that $u_i \in V_u$ and $a_k \in V_a$, and $V_u \cup V_a = V$. The two types of nodes are connected by two types of links (edges): user-to-user links ($e_{uu} = <u_i, u_j>$) and user-to-product links ($e_{ua} = <u_i, a_k>$). Since there is no direct connection for videos in Bilibili, product-to-product links are ignored.

The general aim of network node embedding is to learn a low-dimensional vector representation $\vec{v} \in R^d$ for each node according to links and associated node information. Note that the dimension size $d = |\vec{v}|$ of vector $\vec{v}$ is much smaller than $|V|$, the size of the network.

We propose a novel method to jointly Learn link and Text Embedding for Heterogeneous network nodes (LTEH). Let $\vec{v}^n$ denote link embedding and $\vec{v}^t$ denote text embedding, respectively. Then, the node representation $\vec{v}$ can be obtained by a weighted concatenation between network embedding and text embedding $\vec{v} = \alpha * \vec{v}^n \oplus \beta * \vec{v}^t$, where α and β can be learned though an optimization process.

The objective of LTEH is to obtain optimized node representation by making use of both link and text information. The overall loss function $L(e)$ of all links $e \in E$ is formed by the addition of the network link loss function $L_n(e)$ and the text loss function $L_t(e)$ in a jointly optimized approach defined below:

$$L = \sum_{e \in E} (L_n(e) + L_t(e)). \tag{1}$$

[1] https://www.bilibili.com/

[2] In Bilibili, products are essentially animation videos, hence we use a_k to represent product nodes. In other datasets, nodes can have different names but the process model should be the same.

A. NETWORK EMBEDDING

Since our heterogeneous network contains two types of nodes, the loss function for network embedding $L_n(e)$ in Formula 2 should consider both types of nodes. Let u and a denote user and video nodes, respectively. Since the links of an individual user node can be two types, the e_{uu} type and the e_{ua} type, the loss function of a user should includes two items. On the other hand, a video is only associated with e_{au} links, thus its loss function should have only one item. Consequently, the loss function of network embedding $L_n(e)$, given below in Formula 2, is defined as the addition of two parts: The first part in square brackets is the loss function of user embedding with two link-type probabilities and the second part is the loss function of video embedding with one link-type probability:

$$L_n(e) = \left[W^n_{uu} log(p_n(u_i|u_j)) + W^n_{ua} log(p_n(a_k|u_i))\right] + W^n_{au} log(p_n(u_i|a_k)), \quad (2)$$

where W^n_{uu}, W^n_{ua}, and W^n_{au} are the weight parameter vectors for the three types of links: user-to-user(uu), users-to-(animation) videos (ua), and (animation) video-to-user (au), respectively, the superscript n stand for network embedding. Formula 2 shows that two different types of nodes are represented differently. A user node can connect to both other users and animation videos so its conditional probability has two components defined by the addition of Formula 3 and Formula 4 as the first two elements in Formula 2. Since a video node only has one type of links, its conditional probability is defined by Formula 5 only. The conditional probabilities of the three types of links are listed below:

$$p_n(u_i|u_j) = \frac{exp(\vec{u}_i.\vec{u}_j)}{\sum exp(\vec{u}_i.\vec{V})}, \quad (3)$$

$$p_n(a_k|u_i) = \frac{exp(\vec{a}_k.\vec{u}_i)}{\sum exp(\vec{u}_i.\vec{V})}, \quad (4)$$

and

$$p_n(u_i|a_k) = \frac{exp(\vec{u}_i.\vec{a}_k)}{\sum exp(\vec{a}_k.\vec{V})}, \quad (5)$$

where V refers to all nodes in the network.

B. TEXT BASED OBJECTIVE FUNCTION

The loss function of text embedding should consider text in association with three types of links low in Formula 6, similar to discussion for network embedding:

$$L_t(e) = L_t(u_i, u_j) + L_t(u_i, a_k) + L_t(a_k, u_i). \quad (6)$$

For an e_{uu} link, its loss function is defined by three components:

$$L_t(u_i, u_j) = \alpha_1 L_{tt}(u_i, u_j) + \beta_1 L_{tn}(u_i, u_j) + \gamma_1 L_{nt}(u_i, u_j). \quad (7)$$

In Formula 7, $L_{tt}(u_i, u_j)$ is the loss between text embedding of an e_{uu} link. $L_{tn}(u_i, u_j)$ is the loss between the text embedding of u_i and the network embedding of u_j. $L_{nt}(u_i, u_j)$ is the loss between the network embedding of a user u_i and text embedding of a user node u_j. $\alpha_1, \beta_1, \gamma_1$ are three weighted parameters for the three loss functions. We optimize the conditional probabilities for all the vector representations in Formula 7 as:

$$L_{tt}(u_i, u_j) = W^{tt}_{uu} log(p_{tt}(u_j|u_i)), \quad (8)$$

$$L_{tn}(u_i, u_j) = W^{tn}_{uu} log(p_{tn}(u_j|u_i)), \quad (9)$$

and

$$L_{nt}(u_i, u_j) = W^{nt}_{uu} log(p_{nt}(u_j|u_i)), \quad (10)$$

where W^{tt}_{uu}, W^{tn}_{uu}, and W^{nt}_{uu} are weighted matrices. Similarly, the last two elements in the loss function given in Formula 6 can be defined as:

$$L_t(u_i, a_k) = \alpha_2 L_{tt}(u_i, a_k) + \beta_2 L_{tn}(u_i, a_k) + \gamma_2 L_{nt}(u_i, a_k), \quad (11)$$

and

$$L_t(a_k, u_i) = \alpha_3 L_{tt}(a_k, u_i) + \beta_3 L_{tn}(a_k, u_i) + \gamma_3 L_{nt}(a_k, u_i). \quad (12)$$

$\alpha_2, \beta_2, \gamma_2$ are three heterogeneous weights for the three loss functions in function 11, and $\alpha_3, \beta_3, \gamma_3$ are three weighted parameters for the three loss functions in Formula 12. Probability functions map both link embedding and text embedding onto the same representation space. Softmax function is used to obtain all the probabilities. Now, the main task is to obtain text embeddings of nodes.

C. TEXT EMBEDDING

Most text embedding models examine the context of words at sentence level, which is considered a shallow approach. A more comprehensive approach is to consider the collection of sentences for a node to include other information such as attentions and statistics at a macro level. The main idea in this work treats a collection of sentences in a node as one document to perform embedding in both sentence level and document level. We propose to use a hierarchal neural network to obtain comprehensive semantic information by first capturing aggregated word information at sentence level in one layer, and aggregated sentence information at document level in the second layer.

Let T be the collection of documents associated with n nodes: $T = T_1 \ldots T_i \ldots T_n$. The text T_i for node i is made up by a series of sentences: $T_i = S_1 \ldots S_j \ldots S_{l_i}$ where l_i is the number of sentences in T_i. A sentence S_j is made up of a sequence of words $S_j = w_1{}^j \ldots w_k{}^j, w_{l_j}{}^j$ where l_j is set to be the length of S_j. Each word $w_k{}^j$ is initialized as a fixed dimension vector $\vec{w}^j_k \in R^d$, where d is the size of word vectors.

At the sentence level, sentence embedding using neural networks is learned by three layers:

- Look up layer: Given a word $w_k{}^j \in S_j$, and $S_j \in T_i$, this layer transforms each word into its word embedding.
- Recurrent layer (RNN): Each cell in this layer runs from the first word in the sentence to the last word. For a

sentence S_j in length an l_j, the RNN architecture is an l_j sequential network. The original word vector series $\vec{w}_1^j, \vec{w}_2^j \ldots \vec{w}_n^j$ are transfered to d dimensional hidden vectors through recurrent cells: $\vec{h}_1^j, \vec{h}_2^j \ldots \vec{h}_1^j$. The output is a matrix of size $l_j * d$.

- Average pooling: This layer is used to obtain the embedding of $S_j \in T_i$. Average pooling with non-linear transformation is defined as follows:

$$\vec{S}_j = tanh(avg(\vec{h}_1^j, \vec{h}_2^j \ldots \vec{h}_1^j)). \tag{13}$$

At the document level, our LTEH uses an attention based CNN model. This can give higher weights to the more salient sentences in the collection. The CNN model with attention mechanism consists of three layers:

- Convolution layer: This layer extracts sentence-to-document level information. For a $T_i \in T$ with n sentences $T_i = S_1, S_2 \ldots S_n$, we perform convolution operation over a window of size l by using a convolution matrix $C \in R^{l*(l*d)}$ defined by:

$$\vec{x}_i = C_i S_{i:i+l-1} + b, \tag{14}$$

where $S_{i:i+l-1}$ denotes the concatenation of sentence embeddings learned from the sentence level with window size of l. b is a regularization parameter.

- Attention layer: Attention weights are learned from text contents of both nodes in a link. For an e_{uu} link $<u_i, u_j>$, let the corresponding vector outputs from the convolution layer be $\vec{x}_1^i \ldots \vec{x}_h^i \ldots \vec{x}_n^i$ and $\vec{x}_1^j \ldots \vec{x}_o^j \ldots \vec{x}_m^j$, respectively. The attention weights for each word ${x_o}^i$ and ${x_o}^j$ are defined by:

$$W_{x_h^i} = \frac{\sum_{o=1}^{m} \vec{x}_h^i \vec{x}_o^{j^T}}{Z}, \tag{15}$$

and

$$W_{x_o^j} = \frac{\sum_{h=1}^{n} \vec{x}_o^j \vec{x}_h^{i^T}}{Z}, \tag{16}$$

where

$$Z = \sum_{h=1}^{n} \sum_{o=1}^{m} \vec{x}_h^i \vec{x}_o^{j^T} + \sum_{o=1}^{m} \sum_{h=1}^{n} \vec{x}_o^j \vec{x}_h^{i^T}. \tag{17}$$

The attention weight of word x_h, represented by $W_{x_h^i}$, is calculated from the vector production between its own representation $\vec{x}_h$ and every word in the content of linked user u_j, noted as $\vec{x}_o$. The attention weight of word x_o, represented by $W_{x_o^j}$, is calculated from the production between its own representation $\vec{x}_o$ and every word in the content of linked user u_i, noted as $\vec{x}_h$.

- Pooling: This layer assembles sentence vectors and attention weights into document representations for $\vec{u}_i^t$ and $\vec{u}_j^t$ (the superscript mark t stands for text embedding) as:

$$\vec{u}_i^t = \sum_{h=1}^{n} W_{x_h{}^i} * \vec{x}_h^i, \tag{18}$$

and

$$\vec{u}_i^t = \sum_{o=1}^{m} W_{x_o{}^j} * \vec{x}_o^j. \tag{19}$$

Functions 15, 16, and 17 indicate that in our proposed LTEH model, the attention weight for a sentence S_j is not only determined by its document context, but also text extracted through the linked nodes. Similarly, for e_{ua} links, the attention mechanism can also be obtained. Because user and video have different text content, our model has the ability to capture the differences between the two types of links.

D. TIME COMPLEXITY DISCUSSION

The time complexity of network embedding in our work is basically the same as the LINE model given in Tang's work on large-scale information network embedding (LINE) [23]. Tang's work [23] uses the so called alias table method proposed by Li *et al.* [43] to draw a sample according to the weights of the edges. An alias table method takes only $O(1)$ time when samples are drawn from the same distribution. Based on the same mechanism, network embedding in our proposed method samples an edge from the alias table which takes constant time, $O(1)$. Optimization using negative sampling takes $O(d(K + 1))$ time where K is the number of negative samples and d is the cost of one negative sampling. Therefore, each step takes an overall $O(dK)$ time. In practice, the number of steps used for optimization is usually proportional to the number of edges $O(|E|)$. Therefore, time complexity of network embedding is $O(dK|E|)$, which is linear to the number of edges $|E|$. This result also shows that time complexity of networking embedding is not dependent on the number of nodes $|V|$. Edge sampling method improves the effectiveness of stochastic gradient descent without compromising efficiency.

For text embedding, a two-stage processing architecture is used. For a network with $|V|$ nodes and $|E|$ edges, let us assume that each node has $|l|$ sentence and each sentence has $|m|$ words. Then, the sentence level RNN has a complexity of $O(|m|)$. The document level CNN has a complexity of $O(|l|)$, and attention mechanism has a complexity of $O(|E|)$. Thus, the overall time complexity of text embedding is $O(|m||l|(|V| + |E|))$.

IV. PERFORMANCE EVALUATION

To evaluate the effectiveness of our proposed LTEH model, we conduct link prediction tasks on benchmark datasets. We also use data visualization method to compare our model with CANE-A, the state-of-the-art model.

A. EVALUATED SYSTEMS

Three groups of algorithms are used for performance evaluation and comparison.[3] Group One has three baseline algorithms that only use network link information including:

[3] In the table, three groups are named as G1, G2, G3.

TABLE 1. Statistics of four benchmark datasets.

Data	Nodes	Edges	Words /node	Sent. /node	Total text
Cora	2,277	5,214	90.44	5.32	206K
Hepth	1,038	1,990	54.47	3.26	56K
Zhihu	10,000	43,894	89.61	N/A	896K
Bili_user	3,401	4,259	305.61	12.63	1.039M
Bili_video	1,433	N/A	725.31	29.97	1.039M
Bili_all	4,834	13,801	430.02	17.78	2.079M

- **Deepwalk** [30], a model using local information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences.
- **LINE** [23], a traditional network model using both first-order and second-order proximity in network graph.
- **Node2vec** [14], a random walk based method to sample neighbor nodes.

Group Two algorithms use both link information and text information without the use of attention mechanism including:

- **TADW** [44]: a state-of-the-art algorithm using both network and text information to learn node representation.
- **TriDNR** [45]: a tri-party deep network algorithm that exploits links, node content, as well as label information.
- **CENE** [32]: a method to simultaneously detects community distribution of each node and learns the embeddings of both nodes and communities jointly.
- **CANE-N** [21]: a contrast method by the state-of-the-art CANE system from Tu *et al.*'s work to obtain node embedding through context-aware embedding without attention mechanism.
- **LTEH-N**: a variation of our proposed LTEH, LTEH-N refers to the LTEH model without including the attention mechanism.

Group Three includes two models with built in attention models:

- **CANE-A**: the state-of-the-art attention based network embedding model designed for learning embedding from link and content, proposed by Tu *et al.*
- **LTEH-A**: our full LTEH model with document level processing and attention mechanism [21].

For easy comparison, the network embedding sizes of LTEH-N and LTEH-A are set to 200 as all relevant models in the evaluation dimension size of 200.

B. DATASETS

Table 1 lists the four datasets used for performance evaluation. The datasets are divided into two groups. The first group has three homogeneous networks datasets used in Tu *et al.*'s work [21] with text content including Cora, Hepth, and Zhihu. Details of the three datasets are listed below:

- Cora[4]: A typical citation network dataset [46] consists of 2,708 scientific papers as network nodes (belong to 7 categories) 5,429 links on authors. Cora is the only dataset which has isolated nodes (66 in total) and thus they have no network behavior. This is the first homogeneous network datasets proposed in network embedding.
- Hepth[5] (High Energy Physics Theory): A citation network originally from arXiv3 [47]. This collection is the exact version used by Tu's work [21]. The network has 1,038 paper nodes and 1,990 identical author links.
- Zhihu[6]: An on-line Q&A website in China where users can follow each other and answer questions on this site. The dataset has 10,000 active users and 43,894 links from Zhihu [21]. Punctuations of the text are removed. Thus, there is no document structure.

The second group is a heterogeneous dataset collected by this work for benchmarking from the Bilibili website, referred to as Bilibili. Bilibili is an animation video sharing website of anime, manga and game fandom based in China. Users can submit, view, and add comments on products. In the Bilibili dataset, products are essentially animation videos. Hence, nodes in Bilibili are either user nodes or animation video nodes. The total number of e_{uu} links is 4,259. The total number of e_{ua} links is 9,542. Thus, the total number of links is 13,801. The Bilibili dataset contains 3,400 users and 1,434 animation videos to form a heterogeneous network with a total of 4,834 nodes. The statistics of the four datasets are listed in Table 1.

C. EVALUATION OF LINK PREDICTION

For link prediction, we run the models using the three homogeneous datasets provided by Tu *et al.* [21] in addition to the Bilibili heterogeneous dataset. For fair comparison, we use the reported parameters provided by Tu *et al.* [21] and Pan *et al.* [45] for previous works. Performance is measured by the commonly used AUC values (area under the ROC curve[7]) [48]. Experiments are conducted using different training/test ratios from 15% to 95% with 10% increase in each increment, and average AUC is used on five rounds of random tests. We also show the p-value (the result of t-test) by running our proposed LTEH-A model 10 times and compare it to the state-of-the-art model.[8] In the following tables, results marked by † are performance directly reported by related references. The others are implemented by us [21].

Table 2 shows the results on the two small homogeneous datasets, Cora and Hepth. Table 3 shows the performance of Zhihu. Bold font highlights the best result and the second best is highlighted by underline. Note that CANE-A which uses attention mechanism is indeed the best performer on both Cora and Hepth datasets. Both of models LTEH-N and LTEH-A do not show advantage over the state-of-the-art model in these two relatively small datasets even though LTEH-A does show much improved performance when higher percentage

[4]https://relational.fit.cvut.cz/dataset/CORA

[5]https://snap.stanford.edu/data/cit-HepTh.html

[6]https://www.zhihu.com/

[7]https://en.wikipedia.org/wiki/Receiver-operating-characteristic

[8]All p-values in the Table 2 and 3 is the results of t-test by comparing our proposed LTEH-A model with CANE-A model.

TABLE 2. AUC results of the two small homogeneous datasets Cora and Hepth.

	Model	Dataset	15%	25%	35%	45%	55%	65%	75%	85%	95%
G 1	Deepwalk†	Cora	56.0	63.0	70.2	75.5	80.1	85.2	85.3	87.8	90.3
		Hepth	55.2	66.0	70.0	75.7	81.3	83.3	87.6	88.9	89.0
	LiNE†	Cora	55.0	58.6	66.4	73.0	77.6	82.8	85.6	88.4	89.3
		Hepth	53.7	60.4	66.5	73.9	78.5	83.8	87.5	87.7	87.6
	Node2vec†	Cora	55.9	62.4	66.1	75.0	78.7	81.6	85.9	87.3	88.2
		Hepth	57.1	63.6	69.9	76.2	84.3	87.3	88.4	89.2	89.2
G 2	TADW†	Cora	86.6	88.2	90.2	90.8	90.0	93.0	91.0	93.4	92.7
		Hepth	87.0	89.5	91.8	90.8	91.1	92.6	93.5	91.9	91.7
	TriDNR	Cora	85.1	87.9	88.3	89.1	90.7	90.4	92.2	93.3	94.1
		Hepth	87.5	88.4	89.8	90.4	91.5	91.7	91.9	92.5	92.3
	CENE†	Cora	72.1	86.5	84.6	88.1	89.4	89.2	93.9	95.0	95.9
		Hepth	86.2	84.6	89.8	91.2	92.3	91.8	93.2	92.9	93.2
	CANE-N†	Cora	85.8	90.5	91.6	93.2	93.9	94.6	95.4	95.1	95.5
		Hepth	84.5	89.3	89.2	91.6	91.1	91.8	92.3	92.5	93.6
	LTEH-N†	Cora	82.0	84.9	88.8	89.5	90.3	90.9	91.3	93.7	94.2
		Hepth	86.9	88.0	87.1	90.1	90.0	91.7	91.9	94.1	***95.8***
G 3	CANE-A†	Cora	**86.8**	**91.5**	**92.2**	**93.9**	**94.6**	**94.9**	**95.6**	**96.6**	**97.7**
		Hepth	90.0	91.2	92.0	93.0	94.2	94.6	95.4	95.7	96.3
	LTEH-A	Cora	83.5	86.5	90.4	90.8	92.3	92.9	93.3	94.5	95.4
		Hepth	87.9	88.5	88.9	90.7	90.8	92.9	93.4	96.1	96.8
	(p-value)	Cora	10^{-9}	10^{-11}	10^{-5}	10^{-5}	10^{-13}	10^{-6}	10^{-7}	10^{-5}	0.003
		Hepth	10^{-8}	10^{-8}	10^{-9}	10^{-9}	10^{-10}	10^{-7}	10^{-5}	0.040	0.010

TABLE 3. AUC results of the large homogeneous Zhihu dataset.

	Model	15%	25%	35%	45%	55%	65%	75%	85%	95%
G 1	DeepWalk†	56.6	58.1	60.1	60.0	61.8	61.9	63.3	63.7	67.8
	LINE†	52.3	55.9	59.9	60.9	64.3	66.0	67.7	69.3	71.1
	Node2vec†	54.2	57.1	57.3	58.3	58.7	62.5	66.2	67.6	68.5
G 2	TADW†	52.3	54.2	55.6	57.3	60.8	62.4	65.2	63.8	69.0
	TriDNR	55.1	56.9	61.8	62.3	65.8	68.8	69.2	70.4	71.5
	CENE†	56.2	57.4	60.3	63.0	66.3	66.0	70.2	69.8	73.8
	CANE-N†	56.7	59.1	60.9	64.0	66.1	68.9	69.8	71.0	74.3
	LTEH-N	59.9	62.8	66.3	68.7	69.9	70.6	71.8	72.1	74.8
G 3	CANE-A†	56.8	59.3	62.9	64.5	68.9	70.4	71.4	73.6	75.4
	LTEH-A	**61.9**	**64.8**	**68.3**	**71.9**	**72.5**	**73.4**	**74.2**	**74.5**	**78.9**
	(p-value)	10^{-9}	10^{-9}	10^{-9}	10^{-11}	10^{-13}	10^{-15}	10^{-18}	10^{-21}	10^{-23}

of data can be used. The main reason is that LTEH requires more data for training regardless of homogeneity. On the Cora data, which has 3 % of isolated nodes without links, LTEH again cannot take advantage of its attention model as it needs to follow links to extend the context for attention model to work.

Our proposed LTEH-A starts to show its advantage when training data reaches 75%. This is because LTEH includes a sentence to document level embedding which requires more data for training. For models not using attention mechanism, no particular method has obvious advantage. However, Table 2 indicates that using text information has a clear advantage compared to methods using links only.

Table 3 shows the performance on the Zhihu dataset, which is larger than Cora and Hepth. Note that on the Zhihu dataset LTEH-A has a consistent performance improvement over other baselines including CANE-A. The range of improvement compared to CANE-A is from 0.9% to 7.4%. Table 3 also lists the *p-value*s of our model compared to the best state-of-the-art baseline (CANE-A) to indicate the significance of improvements. As the largest p-value is in the scale of 10^{-9}, this means that improvements in the whole range are very significant. Among all the methods without using attention mechanisms, our proposed LTEH-N also performs better than the state-of-the-art method CANE-N by as much as 5.4%.

Performance on the heterogeneous dataset Bilibili is shown in Table 4. Note that our proposed LTEH-A which uses an attention mechanism consistently outperforms other baselines in all training ratios including the state-of-the-art CANE-A system. The improvement ranges from 0.4% to 5.0%. This indicates that LTEH-A can make more effective use of context information for link prediction. In Bilibili, the largest *p-value* is in the scale of 10^{-9}. This means that improvements by LTEH-A are very significant in the whole data range.

To evaluate the effectiveness of document level embedding, let us now focus on algorithms in the second group. It is interesting to observe that even though Bilibili is rich in text information, none of the methods in this group has a clear advantage. Other than TriDNR, which consistently underperforms in this dataset, the best performers scatter among TADW, CENE, CANE-N, and LTEH-N. By a closer look, LTEH-N gives a good performance with increase margin of at least 0.7% compared to all the other methods

TABLE 4. AUC results of the heterogeneous datasets Bilibili.

	Model	15%	25%	35%	45%	55%	65%	75%	85%	95%
G 1	DeepWalk	51.6	54.1	55.1	55.6	56.8	56.9	57.3	57.7	59.8
	LINE	52.3	55.9	56.9	58.3	59.1	59.4	61.7	62.3	62.8
	Node2vec	54.2	57.1	57.3	58.3	58.7	62.5	66.2	67.6	68.5
G 2	TADW	55.3	56.2	56.6	57.1	60.8	62.4	63.5	64.9	65.8
	TriDNR	55.6	58.7	62.0	62.3	63.7	66.0	68.6	68.6	70.9
	CENE	56.6	59.9	60.4	63.3	64.8	67.9	68.9	69.6	71.0
	CANE-N	57.4	58.7	61.2	62.1	63.3	66.4	67.6	68.0	70.2
	LTEH-N	57.1	57.7	59.3	60.9	62.5	63.9	65.2	68.7	71.1
G 3	CANE-A	57.6	60.7	63.8	64.3	65.0	67.6	69.6	70.5	73.0
	LTEH-A	**60.1**	**62.1**	**64.2**	**66.2**	**68.5**	**69.9**	**71.5**	**75.6**	**78.0**
	(p-value)	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-12}	10^{-9}	10^{-12}	10^{-25}

when the train data reaches 85% and 95%. In other words, even without the use of attention mechanisms, the aggregation of sentence information at document level still helps LTEH-N to outperform CANE-N and CENE when training data is sufficiently large. In Zhihu, LTEH-N outperforms all baseline methods including state-of-the-art CANE-N and CENE in all data range. The improvement range is from 0.5% to 5.4%.

Comparing LTEH-A with LTEH-N, we conclude that the attention mechanism using links and text plays a very important role in node representation. This can be seen from the fact that LTEN-A has achieved higher AUC value in all four datasets than LTEN-N. Obviously, richer text information with extended context helps to built a better attention model. In the Bilibili dataset, a node has far more number of words and sentences than that in the other three datasets. On average, LTEN-A is 2.0% to 6.9% higher in AUC in Bilibili (Table 4) than LTEN-N, compared to about 1.0 % improvement in Hepth and Cora (Table 2).

D. EVALUATION OF DIFFERENT HETEROGENEOUS WEIGHTS

In each formula of 7, 11, and 12, three parameters are used as weighted parameters. Firstly we evaluate the three weighted parameters in formula 7, $L_{tt}(u_i, u_j)$ is the loss between text embedding of an e_{uu} link. $L_{tn}(u_i, u_j)$ is the loss between the text embedding of u_i and the network embedding of u_j. $L_{nt}(u_i, u_j)$ is the loss between the network embedding of a user u_i and text embedding of a user node u_j. $\alpha_1, \beta_1, \gamma_1$ are three weighted parameters for the three loss functions. When we evaluate the effect of α_1 in formula 7, we take the Bilibili dataset using 85% of nodes and links and the value of α_1 ranges from 0.1 to 1 with the increment of 0.1 in each step, while the other two parameters are fixed to 1. We use the same process to evaluate the effect of two other parameters β_1, γ_1 in function 7. The evaluation process of other parameters in formula 11, and 12 are the same as formula 7.

Figure 1 shows the effect of three parameters in formula 7, we observe that α_1 affects the performance dramatically. Decreasing value of α_1 has negative impact. On the other hand, the change of β_1 and γ_1 will have much less effect on the performance of LTEH-A model. Because increasing

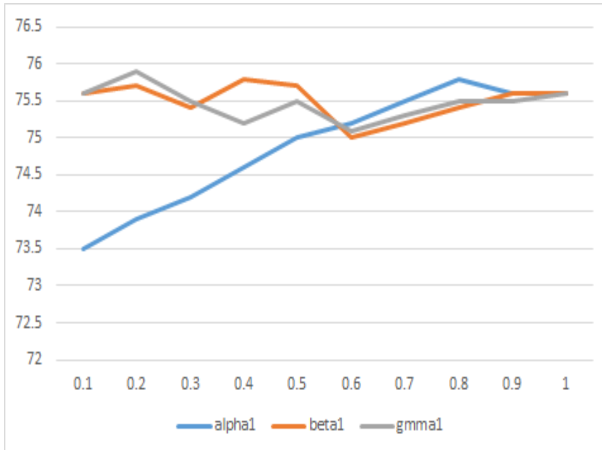


FIGURE 1. Evaluation of three parameters in formula 7.

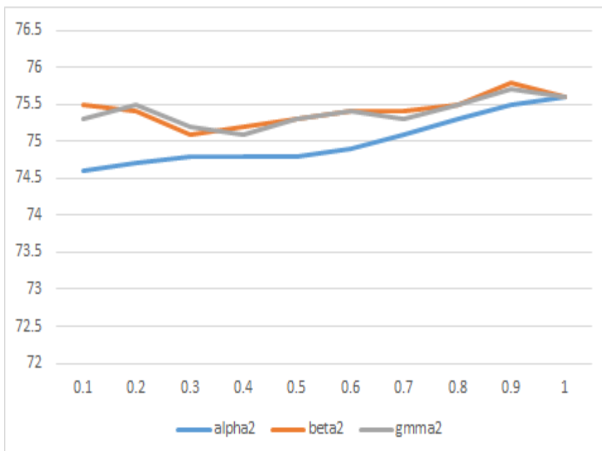


FIGURE 2. Evaluation of three parameters in formula 11.

value of α_1 essentially means increasing the significance of text embedding, which indicates that text embedding plays an important role in encoding the user to user relationship.

Changing the weight of parameters in formula 11, and formula 12 has much less impact on the performance of LTEH-A model. But we still can observe that the increasing of α_2, α_3 can improve the performance of LTEH-A model, while β_2, β_3 and γ_2, γ_3 have relatively small effect on the

TABLE 5. AUC of Local-Text based attention mechanism vs. Text&Link based attention mechanism in LTEH.

	Model	15%	25%	35%	45%	55%	65%	75%	85%	95%
Cora	Local-Text	82.1	83.5	84.4	86.6	90.2	91.8	92.0	93.1	94.4
	Text&Link	**83.5**	**86.5**	**90.4**	**86.5**	**92.3**	**92.9**	**93.3**	**94.5**	**95.4**
Hepth	Local-Text	85.1	86.9	87.6	87.9	88.8	90.2	91.7	93.4	95.5
	Text&Link	**87.9**	**88.5**	**88.9**	**90.7**	**90.8**	**92.9**	**93.4**	**96.1**	**96.8**
Zhihu	Local-Text	59.8	62.3	65.9	69.5	68.9	70.4	72.4	73.1	76.4
	Text&Link	**61.9**	**64.8**	**68.3**	**71.9**	**72.5**	**73.4**	**74.2**	**74.5**	**78.9**
Bilibili	Local-Text	58.8	59.3	61.4	64.5	68.9	69.0	70.7	74.1	76.4
	Text&Link	**60.1**	**62.1**	**64.2**	**66.2**	**68.5**	**69.9**	**71.5**	**75.6**	**78.0**

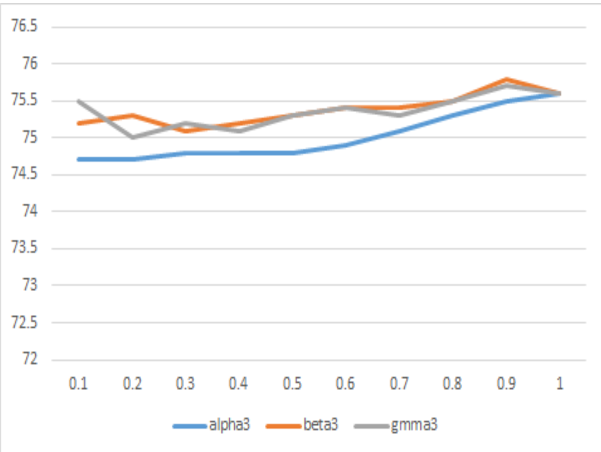


FIGURE 3. Evaluation of three parameters in formula 12.

performance of LTEH-A model. The evaluations of three parameters in three functions indicate that text information of a link encodes valuable information in network embedding.

E. EVALUATION OF ATTENTION MECHANISM

To examine the effectiveness of the attention mechanism using link extensions in our proposed attention mechanism, we split our attention mechanism without changing any other part of the algorithm by using (1) only local text as context for attention, labeled as **Local-Text** and (2) local text with extended context using both local text and extended text by links, labeled as **Text&Links** by our model. Table 5 gives the performance of two different attention mechanisms in link prediction task measured by AUC. The improvements in all the four datasets are clear and substantial. In Zhihu and Bilibili, text information is far richer than Cora and Hepth. For Zhihu, Text&Link achieves 2.50% net increase on average in AUC compared to Local-Text in different training proportions; in Bilibili, the average net increase is 1.73%. The average net increases reach 2.02% in Cora and 2.11% in Hepth. The result of this part proves that incorporating link information into attention mechanism can provide more information about node characteristics.

F. VISUALIZATION

Performance evaluation can also be observed by producing a visualization of a network in two-dimensional space. Visualizations can help to understand network topology. Visualization in our work is performed on the node

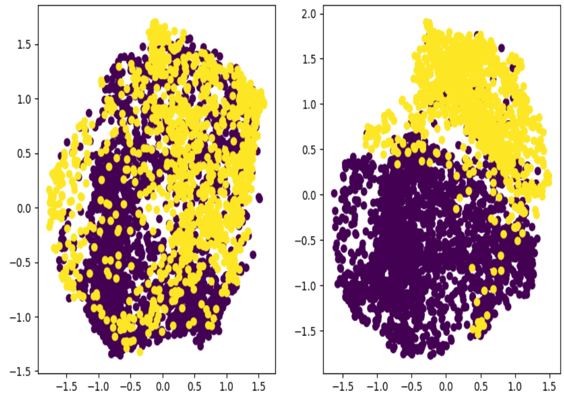


FIGURE 4. Visualization of two node types on Bilibili dataset(Left:CANE, Right:LTEH-A).

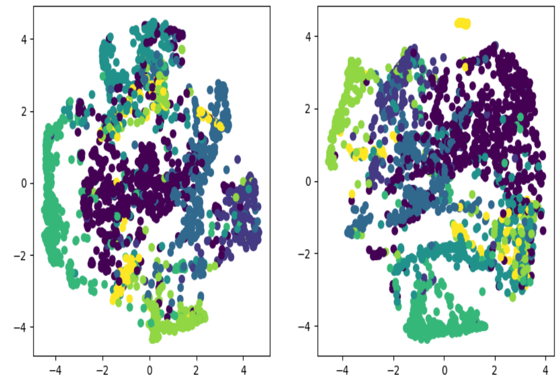


FIGURE 5. Visualization of seven types of user nodes in Cora dataset (Left:CANE, Right:LTEH-A).

representation of a 200 dimension node vector by the t-SNE algorithm [49][9] to reduce the dimensionality to 2.

Figure 4 shows the visualization of node types for the Bilibili dataset with the state-of-the-art system CANE-A on the left and our proposed LTEH-A on the right in the training ratio of 0.95. The yellow color represents the video nodes and the purple represents user nodes. It is easy to see that LTEH-A can separate the two types of nodes much better.

Figure 5 shows the visualization results of CANE-A and LTEH-A in Cora by reducing the dimensions to 7 groups. Even though Cora is a homogeneous network with only authors, each author belongs to one of 7 different categories of authors. Visualization result shows that our model LTEH-A on the right still makes comparable result to the state-of-the-art CANE-A on the left.

[9]https://lvdmaaten.github.io/tsne/

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel model to learn node embedding for heterogeneous networks through a joint learning framework of both network links and text associated with nodes. The novelty of our proposed model includes two parts. Firstly, we learn the embedding of different nodes separately from links and other types of contents. Hence our model is capable of learning different types of nodes in heterogeneous networks. Secondly, we propose a novel attention mechanism to extend text by following links of adjacent nodes such that much larger context of the network can be included.

Experiment shows that our proposed LTEH-A outperforms the current state-of-the-art network embedding methods. The net increase in performance in a heterogeneous network ranges from 2.5% to 5.0 % for link prediction. Evaluation shows that the attention mechanism is a very important element to obtain important features for network node representation as richer text information with extended context helps to built a better attention model. Both evaluation tasks show that our proposed joint learning framework and attention mechanism can learn node representation more effectively.

In general, node representation should be able to learn from all kinds of associated information of nodes besides links. In this work, however, we only used text data linked to nodes. As a direction for future work, other information such as images and animation videos can also be explored.

ACKNOWLEDGMENT

(Yunfei Long and Rong Xiang contributed equally to this work.)

REFERENCES

[1] G. Amitai *et al.*, "Network analysis of protein structures identifies functional residues," *J. Mol. Biol.*, vol. 344, no. 4, pp. 1135–1146, 2004.
[2] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, "Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding," *Bioinformatics*, vol. 29, no. 13, pp. i199–i209, 2013.
[3] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PloS one*, vol. 10, no. 11, p. e0141287, 2015.
[4] R. F. I. Cancho and R. V. Solé, "The small world of human language," *Proc. Roy. Soc. London B, Biol. Sci.*, vol. 268, no. 1482, pp. 2261–2265, 2001.
[5] L. C. Freeman, "Visualizing social networks," *J. Social Struct.*, vol. 1, no. 1, p. 4, 2000.
[6] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1225–1234.
[7] P. Goyal and E. Ferrara. (2017). "Graph embedding techniques, applications, and performance: A survey." [Online]. Available: https://arxiv.org/abs/1705.02801
[8] B. Guo, D. Zhang, Z. Yu, Y. Liang, Z. Wang, and X. Zhou, "From the Internet of Things to embedded intelligence," *World Wide Web*, vol. 16, no. 4, pp. 399–420, 2013.
[9] Z. Yu, H. Xu, Z. Yang, and B. Guo, "Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 151–158, Feb. 2016.
[10] R. F. I. Cancho and R. V. Solé, "The small world of human language," *Proc. Roy. Soc. London B, Biol. Sci.*, vol. 268, no. 1482, pp. 2261–2265, 2001.
[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, vol. 14, 2014, pp. 1532–1543.
[13] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1165–1174.
[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.
[15] K.-W. Chang, M.-W. Chang, A. Rush, and V. Srikumar, "Proceedings of the workshop on structured prediction for NLP," in *Proc. Workshop Struct. Predict. NLP*, 2016, pp. 1–71.
[16] W. Ding, C. Lin, and P. Ishwar, "Node embedding via word embedding for network community discovery," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 3, pp. 539–552, Sep. 2017.
[17] K. Tu, P. Cui, X. Wang, F. Wang, and W. Zhu, "Structural deep embedding for hyper-networks," in *Proc. AAAI*, 2018, pp. 426–433.
[18] M. Ou, P. Cui, F. Wang, J. Wang, W. Zhu, and S. Yang, "Comparing apples to oranges: A scalable solution with heterogeneous hashing," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 230–238.
[19] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, to be published, doi: 10.1109/TKDE.2018.2849727.
[20] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 263–272.
[21] C. Tu, H. Liu, Z. Liu, and M. Sun, "Cane: Context-aware network embedding for relation modeling," in *Proc. 55th Annu. Meeting ACL*, 2017, pp. 1722–1731.
[22] M. Jiang *et al.*, "Social contextual recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 45–54.
[23] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
[24] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[25] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, Sep. 2008.
[26] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, "Distributed large-scale natural graph factorization," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 37–48.
[27] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.
[28] M. E. J. Newman, "A measure of betweenness centrality based on random walks," *Soc. Netw.*, vol. 27, no. 1, pp. 39–54, 2005.
[29] 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. Knowl. Data Eng.*, vol. 19, no. 3, pp. 355–369, Mar. 2007.
[30] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.
[31] Y. Long, Q. Lu, Y. Xiao, M. Li, and C.-R. Huang, "Domain-specific user preference prediction based on multiple user activities," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2016, pp. 3913–3921.
[32] C. Tu, H. Wang, X. Zeng, Z. Liu, and M. Sun. (2016). "Community-enhanced network representation learning for network analysis." [Online]. Available: https://arxiv.org/abs/1611.06645
[33] L. Gui, R. Xu, Y. He, Q. Lu, and Z. Wei, "Intersubjectivity and sentiment: From language to knowledge," in *Proc. IJCAI*, 2016, pp. 2789–2795.
[34] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 119–128.
[35] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 151–161.
[36] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2013, pp. 1631–1642.

[37] O. Irsoy and C. Cardie, "Opinion mining with deep recurrent neural networks," in *Proc. EMNLP*, 2014, pp. 720–728.

[38] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[39] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1422–1432.

[40] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics and 7th Int. Joint Conf. Natural Lang. Process.*, Beijing, China, vol. 1, Jul. 2015, pp. 1014–1023.

[41] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. HLT-NAACL*, 2016, pp. 1480–1489.

[42] W. Yin, H. Schütze, B. Xiang, and B. Zhou. (2015). "ABCNN: Attention-based convolutional neural network for modeling sentence pairs." [Online]. Available: https://arxiv.org/abs/1512.05193

[43] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, "Reducing the sampling complexity of topic models," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 891–900.

[44] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proc. IJCAI*, 2015, pp. 2111–2117.

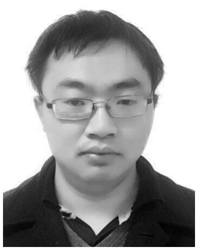
[45] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," *Network*, vol. 11, no. 9, p. 12, 2016.

[46] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of Internet portals with machine learning," *Inf. Retr.*, vol. 3, no. 2, pp. 127–163, 2000.

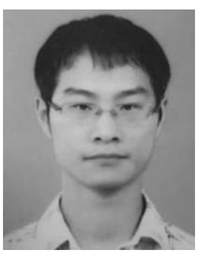
[47] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 177–187.

[48] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[49] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.



YUNFEI LONG received the double bachelor's degree in computer science and linguistics from Jilin University, Changchun, China, in 2013, and the M.Sc. degree in cognitive science from The University of Edinburgh, U.K., in 2015. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University. His current research interests include natural language processing, neural network, and social media analysis.



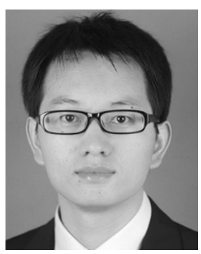
RONG XIANG received the bachelor's degree in electronic and information engineering from the Wuhan University of Technology, Wuhan, China, in 2010, the M.Sc. degree in information engineering from Zhejiang University, Hangzhou, China, in 2013, and the M.Sc. degree in cognitive science from The University of Edinburgh, U.K., in 2015. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University. From 2015 to 2017, he was a Software Engineer with Advanced Micro devices Inc., Shanghai, China. His current research interests include natural language processing, neural network, and knowledge base.

QIN LU is currently a Professor with The Hong Kong Polytechnic University. Her main research works include computational linguistics, i.e., using computational methods to process Chinese text, extract useful information, and build Chinese NLP-related resources. Her expertise is in lexical semantics, text mining, opinion analysis, and knowledge discovery.

DAN XIONG is currently a Research Staff working with Prof. Q. Lu with The Hong Kong Polytechnic University. Her research interests are mainly in lexical semantics, NLP-related language resources, and Chinese character standardization and encoding.

CHU-REN HUANG received the Ph.D. degree in linguistics from Cornell University in 1987. He is currently the Dean of the Faculty of Humanities and the Chair Professor of applied Chinese language studies with The Hong Kong Polytechnic University. He has played a central role in developing Chinese language resources and in leading the fields of Chinese corpus and computational linguistics.

CHENGLIN BI received the bachelor's degree in electronic and information engineering from Shanghai Second Polytechnic University, Shanghai, China, in 2011. Since 2013, he has been a Software Engineer with Advanced Micro devices (Shanghai), where he is involved in GPU performance tuning. His current research interests include CV, natural language processing, and neural network.



MINGLEI LI received the B.E. degree in mechanical engineering and the M.E. degree in mechanical and electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2011 and 2014, respectively, and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2018. He is currently a Researcher with Huawei Technologies Co., Ltd. His research interests include natural language processing, sentiment and emotion analysis, computational linguistic, and applied machine learning.