Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks

Yu Shi* Qi Zhu* Fang Guo Chao Zhang Jiawei Han
University of Illinois at Urbana-Champaign, Urbana, IL USA
{yushi2, qiz3, fangguo1, czhang82, hanj}@illinois.edu

ABSTRACT

Heterogeneous information networks (HINs) are ubiquitous in real-world applications. In the meantime, network embedding has emerged as a convenient tool to mine and learn from networked data. As a result, it is of interest to develop HIN embedding methods. However, the heterogeneity in HINs introduces not only rich information but also potentially incompatible semantics, which poses special challenges to embedding learning in HINs. With the intention to preserve the rich yet potentially incompatible information in HIN embedding, we propose to study the problem of comprehensive transcription of heterogeneous information networks. The comprehensive transcription of HINs also provides an easy-to-use approach to unleash the power of HINs, since it requires no additional supervision, expertise, or feature engineering. To cope with the challenges in the comprehensive transcription of HINs, we propose the HEER algorithm, which embeds HINs via edge representations that are further coupled with properly-learned heterogeneous metrics. To corroborate the efficacy of HEER, we conducted experiments on two large-scale real-words datasets with an edge reconstruction task and multiple case studies. Experiment results demonstrate the effectiveness of the proposed HEER model and the utility of edge representations and heterogeneous metrics. The code and data are available at https://github.com/GentleZhu/HEER.

CCS CONCEPTS

•**Information systems** → **Data mining;** •**Computing methodologies** → **Learning latent representations;** *Neural networks;*

KEYWORDS

Heterogeneous information networks, network embedding, graph mining, representation learning.

ACM Reference format:
Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. 2018. Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, United Kingdom, August 19–23, 2018 (KDD '18).* DOI: 10.1145/3219819.3220006

*These authors contributed equally to this work.

KDD '18, August 19–23, 2018, London, United Kingdom

978-1-4503-5552-0/18/08…$15.00
DOI: http://dx.doi.org/10.1145/3219819.3220006

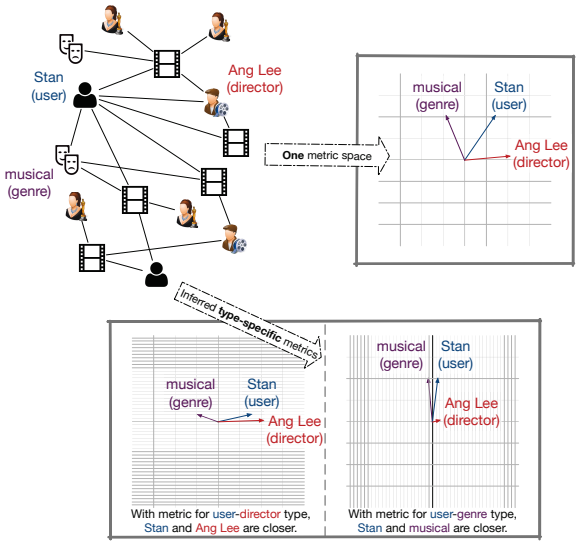


Figure 1: To preserve the rich information in HIN embedding, properly handling the incompatibility introduced by the heterogeneity is necessary. The upper left part of the figure gives a toy movie-reviewing HIN, where users review movies and list certain directors, actors, genres as their favorites. *Stan* likes both *musical* and movies directed by *Ang Lee*. If all nodes were embedded to one metric space, *Stan* would be close to neither *musical* nor *Ang Lee* due to the dissimilarity between *musical* and *Ang Lee*. This results in information loss in the embedding learning process. However, we can alleviate this problem by employing edge representation and inferring edge-type–specific metrics, so that *Stan* can be close to both *musical* and *Ang Lee* under their respective metrics, while not necessarily dragging *musical* and *Ang Lee* closer. The two metrics shown in the lower figure can be achieved by linearly transforming the metric space in the upper right figure.

1 INTRODUCTION

Heterogeneous information networks (HINs) have received increasing attention in the past decade due to its ubiquity and capability of representing rich information [21, 25]. Meanwhile, network embedding has emerged as a scalable representation learning method [5, 7, 18, 19, 28, 29, 32]. Network embedding learns low-dimensional vector representations for nodes to encode their semantic information in the original network. The vectorized representations can be easily combined with off-the-shelf machine learning algorithms for various tasks such as classification and link prediction [7, 9, 18, 29], which provides a convenient approach for researchers and engineers to mine and learn from the networked data. To marry the advantages of HINs and network embedding,

researchers have recently started to explore methods to embed heterogeneous information networks [2, 5, 6, 8, 20, 23, 28], and have demonstrated the effectiveness of HIN embedding in applications including author identification [3], name disambiguation [35], proximity search [11], event detection [36], *etc.*

However, the heterogeneity in HINs brings in not only rich information but also potentially incompatible semantics, which poses special challenges to embed heterogeneous information networks. Take the movie-reviewing network in Figure 1 as an example, where users review movies and list certain actors, directors, and genres as their favorites. Suppose user *Stan* likes both movies directed by *Ang Lee* (director) and *musical* (genre). Since *Ang Lee* has never directed any *musical*, nor is he semantically similar to *musical*, if this HIN were embedded into *one* metric space, *musical* and *Ang Lee* would be distant from each other, while the user *Stan* would not be simultaneously close to both of them, due to the triangle inequality property of metric spaces. We have also observed different extents of such incompatibility from real-world data as to be discussed in Section 4, which is consistent with the observation that different extents of correlation can exist within one HIN as per existing study [22]. As a result, it can be expected that an algorithm would generate better embeddings if it additionally models such semantic incompatibility. We hence study the problem of *comprehensive transcription of heterogeneous information networks*, which purely aims to transcribe the rich and potentially incompatible information from HINs to the embeddings, without involving additional expertise, feature engineering, or installation of supervision.

With HINs comprehensives transcribed, one can again pipe the *unsupervisedly* learned embeddings to off-the-shelf machine learning algorithms for a wide range of applications. Therefore, beyond the capability of preserving rich information, another motivation to study comprehensive transcription of HINs is to provide an easy-to-use approach to unleash the power of HINs in a wide variety of applications with no expertise or supervision required in the embedding learning process.

Traditional homogeneous network embedding methods [7, 18, 19, 29, 32] treat all the nodes and edges equally regardless of their types, which do not capture the essential heterogeneity of HINs. A couple of methods have recently been studied for embedding heterogeneous information networks [2, 5, 6, 8, 20, 23, 28]. Many of them build their algorithms on top of a set of meta-paths [5, 20], which often require users to specify the meta-paths or leverage supervision to make the meta-path selection. However, a set of meta-paths specified or selected in this way often only reflects certain aspects of the HIN or is suitable for specific tasks. As a result, they are not always capable of transcribing HINs comprehensively. These methods are not as easy-to-use either because it involves the additional meta-path generation process that entails expertise or supervision. Besides using meta-paths, some approaches have been proposed to embed specific kinds of HINs [8, 28] for certain tasks or HINs with additional side information [2]. These methods cannot be applied to comprehensively transcribe general HINs. Additionally, most existing HIN embedding methods [5, 8, 20, 28] employ only one metric space for embedding learning. This approach may suit downstream tasks that are related to certain partial information of an HIN with compatible semantics but could lead to information loss if the objective is to comprehensively transcript the entire HIN.

The problem of comprehensive transcription of HINs is challenging because it requires the modeling of heterogeneity that can be complex and incompatible. Besides, without the availability of supervision, proposed solutions need to capture the latent structure of the HINs and distinguish potentially incompatible semantics in an unsupervised way. To cope with these challenges, we propose **h**eterogeneous information network **e**mbedding via **e**dge **r**epresentations, which is henceforth referred to as HEER. HEER builds edge embeddings atop node embeddings, which are further coupled with inferred heterogeneous metrics for each edge type. The inferred metrics capture which dimensions of the edge embeddings are more important for the semantic carried by their corresponding edge types. In turn, the information carried by edges of different types updates the node embeddings and edge embeddings with emphases on different type-specific manifolds. In this way, we can preserve different semantics even in the presence of incompatibility. Still take the movie-reviewing network as example, by adopting heterogeneous metrics as in the lower part of Figure 1, *Stan* could be close to both *musical*(genre) and *Ang Lee*(director) under their respective metrics. Furthermore, the heterogeneous metrics are inferred by fitting the input HIN, so that semantic incompatibility is captured without additional supervision.

Specifically, with the availability of edge representations and coupled metrics, we derive loss function that reflects both the existence and the type of an edge. By minimizing the loss, the node embeddings, edge embeddings, and heterogeneous metrics are updated simultaneously, and thereby retain the heterogeneity in the input HIN. Different extents of incompatibility can also be modeled, where the more compatible two edge types are, the more similar their corresponding metrics would be.

Lastly, we summarize our contributions as follows:

(1) We propose to study the problem of comprehensive transcription of HINs in embedding learning, which preserves the rich information in HINs and provides an easy-to-use approach to unleash the power of HINs.
(2) We identify that different extents of semantic incompatibility exist in real-world HINs, which pose challenges to the comprehensive transcription of HINs.
(3) We propose an algorithm, HEER, for the comprehensive transcription of HINs that leverages edge representations and heterogeneous metrics.
(4) Experiments with real-world large-scale datasets demonstrate the effectiveness of HEER and the utility of edge representations and heterogeneous metrics.

2 RELATED WORK

Homogeneous network embedding. Meanwhile, network embedding has emerged as an efficient and effective representation learning approach for networked data [4, 7, 9, 16, 18, 19, 19, 29, 32, 37], which significantly spares the labor and sources in transforming networks into features that are more machine-actionable. Early network embedding algorithms start from handling the simple, homogeneous networks, and many of them trace to the skip-gram model [13] that aims to learn word representations where words with similar context have similar representation [7, 18, 19, 29].

Besides skip-gram, algorithms for preserving certain other homogeneous network properties have also been studied [10, 15, 16, 31–33]. The use of edge representations for homogeneous network embedding is discussed in a recent work [1], but such edge representations are designed to distinguish the direction of an edge, instead of encoding richer semantics such as edge type in our case.

Heterogeneous network embedding. Heterogeneous information network (HIN) has been extensively studied since the past decade for its ubiquity in real-world data and efficacy in fulfilling tasks, such as classification, clustering, recommendation, and outlier detection [21, 25, 27, 34, 38]. To marry the advantages of HIN and network embedding, a couple of algorithms have been proposed very recently for embedding learning in heterogeneous information networks [2, 5, 6, 8, 20, 23, 28]. One line of work first uses human expertise or supervision to select meta-paths for a given task or limit the scope of candidate meta-paths, and then proposes methods to transfer the semantics encoded in meta-paths to the learned embedding [5, 6, 20]. While this direction has been showed to be effective in solving problems that fit the semantics of the chosen meta-paths, it differs from the research scope of ours because they mostly focus on providing quality representations for downstream tasks concerning the node types on the two ends of chosen meta-paths, while we aim at developing methods to transcribe the entire HIN to embeddings as comprehensively as possible. Beyond meta-paths, some approaches have been proposed to embed specific kinds of HINs [8, 28] with specific objectives such as representing event data or learning predictive text embeddings. Some other approaches study HINs with additional side information [2] that cannot be generalized to all HINs. Besides, all of these approaches embed the input HIN into only one metric space. Embedding in the context of HIN has also been studied for tasks with additional supervision [3, 11, 17]. These methods either yield features specific to given tasks, and are outside of the scope of unsupervised HIN embedding that we study.

A recent study [23] proposes a method by decomposing an HIN into multiple aspects before learning embedding, which also attains quality representations of HINs by alleviating the information loss arising from the rich, yet heterogeneous, and potentially conflicting semantics within the given networks. However, this approach embeds the derived aspects independently and completely forbids joint learning across aspects while our proposed method allows network components of varied compatibility to collaborate to different extents in the joint learning process.

3 PRELIMINARIES

In this section, we define related concepts and notations.

Definition 3.1 (Heterogeneous Information Network). An **information network** is a directed graph $G = (\mathcal{V}, \mathcal{E})$ with a node type mapping $\varphi : \mathcal{V} \to \mathcal{T}$ and an edge type mapping $\psi : \mathcal{E} \to \mathcal{R}$. Particularly, when the number of node types $|\mathcal{T}| > 1$ or the number of edge types $|\mathcal{R}| > 1$, the network is called a **heterogeneous information network** (HIN).

Given the typed essence of HINs, the network schema $\tilde{G} = (\mathcal{T}, \mathcal{R})$ [25] is used to abstract the meta-information regarding the node types and edge types in an HIN. Figure 2 illustrates the schema of a toy movie-reviewing HIN.

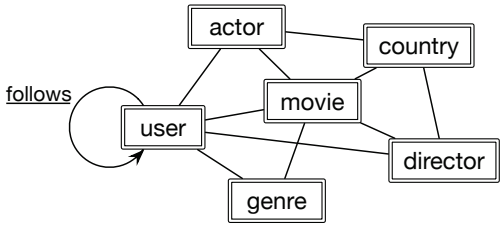


Figure 2: The schema of a toy movie-reviewing HIN with six node types, seven undirected edge types, and one directed edge type.

In addition, we require that only one node type can be associated with a certain end of an edge type. That is, once an edge type is given, we would deterministically know the node types on its two ends. As an example, consider two edges with one representing director *Fatih Akin* living in *Germany* and another representing movie *In the Fade* being produced in *Germany*. Such requirement implies that these two edges must have distinct types – *livesIn* and *isProducedIn* – instead of just one type – *isIn*. For edge type $r \in \mathcal{R}$, we denote $\mathcal{P}^r := \left\{(u, v) \in \mathcal{V} \times \mathcal{V} \,\middle|\, \varphi(u) \sim r \sim \varphi(v)\right\}$, where $\varphi(u) \sim r \sim \varphi(v)$ means the node type pair $(\varphi(u), \varphi(v))$ is consistent with edge type r. Additionally, define $\mathcal{P}^r_{u*} := \left\{\tilde{v} \in \mathcal{V} \middle| (u, \tilde{v}) \in \mathcal{P}^r\right\}$ and $\mathcal{P}^r_{*v} := \left\{\tilde{u} \in \mathcal{V} \middle| (\tilde{u}, v) \in \mathcal{P}^r\right\}$.

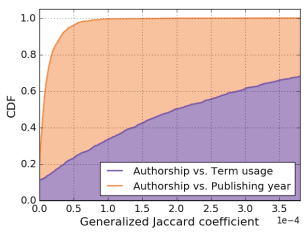
Moreover, when the network is weighted and directed, we use $W^{(r)}_{uv}$ to denote the weight of an edge $e \in \mathcal{E}$ with type $r \in \mathcal{R}$ that goes out from node u toward v. $D^{O(r)}_u$ and $D^{I(r)}_u$ respectively represent the outward degree of node u (*i.e.*, the sum of weights of all type-r edges going outward from u) and the inward degree of node u (*i.e.*, the sum of weights of all type-r edges going inward to u). For an unweighted edge, $W^{(r)}_{uv}$ is trivially 1. For an undirected edge, we always have $W^{(r)}_{uv} = W^{(r)}_{vu}$ and $D^{O(r)}_u = D^{I(r)}_u$.

Definition 3.2 (Node and Edge Representations in HIN Embedding). Given an HIN $G = (\mathcal{V}, \mathcal{E}; \varphi, \psi)$, the problem of HIN embedding via edge representations learns a node embedding mapping $f : \mathcal{V} \to \mathbb{R}^{d_\mathcal{V}}$ and an edge embedding mapping $f : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^{d_\mathcal{E}}$, where $d_\mathcal{V}$ and $d_\mathcal{E}$ are the dimensions for node and edge embeddings, respectively. A node $u \in \mathcal{V}$ is thereby represented by a node embedding $\mathbf{f}_u := f(u)$ and a node pair $(u, v) \in \mathcal{V} \times \mathcal{V}$ is represented by an edge embedding $\mathbf{g}_{uv} := g(u, v)$.

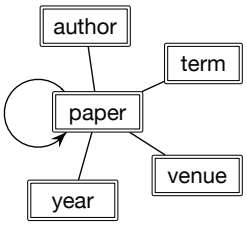
With this definition, a node pair has its edge embedding even if no edge of any type has been observed between them. On the other hand, it is possible for node pair (u, v) to be associated by multiple edges with different types, and we expect edge embedding $\mathbf{g}_{uv}$ to encapsulate such information of an HIN.

Finally, we define the problem of comprehensive transcription of a heterogeneous information network in embedding learning.

Definition 3.3 (Comprehensive Transcription of an HIN). The comprehensive transcription of an HIN aims to learn the representations of the input HIN that retains the rich in the HIN as comprehensively as possible, in an approach that does not require additional expertise, feature engineering, or supervision.



(a) The CDF of the generalized jaccard coefficients for two pairs of edge types.



(b) The schema of the DBLP network.

Figure 3: Varied extents of incompatibility exist between different pairs of edge types in the DBLP network.

4 VARIED EXTENTS OF INCOMPATIBILITY DUE TO HETEROGENEITY

In this section, we look into the incompatibility in HINs using real-world data, and we take DBLP as an example.

DBLP is a bibliographical information network in the computer science domain [30], where authors write papers that are further associated with nodes of other attribute types. Since the measurable incompatibility in an HIN arises from the co-existence of multiple edge types, we dive down to the minimal case that involves two different edge types (r_1 and r_2) joined by a common node type (t). To quantify the incompatibility for this minimal case, we use the widely used generalized Jaccard coefficient to measure the similarity between the node groups reachable from a given node of type t via the two edge types. Specifically, given node u of type t, the Jaccard coefficient for edge types r_1 and r_2 is given by $J(u; r_1, r_2) := \frac{\min_{\varphi(v)=t}\{l(u,v;r_1), l(u,v;r_2)\}}{\max_{\varphi(v)=t}\{l(u,v;r_1), l(u,v;r_2)\}}$, where $l(u,v;r) := \boldsymbol{P}^r_{u,:}(\boldsymbol{P}^r_{v,:})^\top$ is the reachability between nodes u and v via edge type r and $\boldsymbol{P}^r$ is the row-normalized adjacency matrix of edge type r. Generalized Jaccard coefficient has a range of $[0, 1]$, and greater value implies more similarity, or equivalently, less incompatibility.

As an example, we consider four node types – author, paper, key term, and year – and two pairs of edge types – (i) authorship vs. publishing year of papers and (ii) authorship vs. term usage of papers. We illustrate the distributions over Jaccard coefficient using cumulative distribution function (CDF) for each of the two pairs in Figure 3a. It can be seen that over 95% of nodes have a generalized Jaccard coefficient smaller than $5e^{-5}$ between authorship and publishing year, while less than 25% of nodes fall in the same category when it comes to authorship vs. term usage. In other words, we observe more incompatibility between authorship and publishing year than between authorship and term usage. However, this relationship is actually not surprising because papers published in the same year can be authored by any researchers who are active at that time, while key terms associated to certain research topics are usually used by authors focusing on these topics. With the presence of such varied extent of incompatibility, we would expect an embedding algorithm tailored for comprehensive transcription of HINs to be able to capture this semantic subtlety in HINs.

In fact, by employing edge representation and heterogeneous metrics, the inferred metrics could be learned to be different for incompatible edge types. In turn, the information carried by these two edge types would be updating the node embeddings and edge embeddings with emphases on different manifolds. On the other hand, the subtlety of the different extent of incompatibility could also be captured in a way that the more compatible two edge types are, the more similar their inferred metrics should be.

5 PROPOSED METHOD

To provide an general-purpose, easy-to-use solution to HIN embedding, we describe the HEER model in this section, where HEER stands for **H**eterogeneous Information Network **E**mbedding via **E**dge **R**epresentations. Afterward, the model inference method is described subsequently.

5.1 The HEER Model

A learned embedding that effectively encodes the semantics of an HIN should be able to reconstruct this HIN. With the use of edge representation, we expect the embedding to infer not only the existence but also the type of edge between each pair of nodes. For edge type $r \in \mathcal{R}$, we formulate the **typed closeness** of node pair (u, v) atop their edge embedding $\mathbf{g}_{uv}$ as

$$s_r(u,v) := \begin{cases} \dfrac{\exp(\boldsymbol{\mu}_r^\top \mathbf{g}_{uv})}{\sum_{\tilde{v}\in\mathcal{P}^r_{u*}} \exp(\boldsymbol{\mu}_r^\top \mathbf{g}_{u\tilde{v}}) + \sum_{\tilde{u}\in\mathcal{P}^r_{*v}} \exp(\boldsymbol{\mu}_r^\top \mathbf{g}_{\tilde{u}v})}, & (u,v) \in \mathcal{P}^r, \\ 0, & (u,v) \notin \mathcal{P}^r, \end{cases} \tag{1}$$

where $\boldsymbol{\mu}_r \in \mathbb{R}^{d_\mathcal{V}}$ is a edge-type–specific vector to be inferred that represents the metric coupled with this type. Edge types with compatible semantics are expected to share similar $\boldsymbol{\mu}_r$, while incompatible edge types make use of different $\boldsymbol{\mu}_r$ to avoid the embedding learning on respective semantics to dampen each other.

To measure the capability of the learned embedding in reconstructing the input HIN, the difference between the observed weights and the typed closeness inferred from embedding are used, which leads to the objective to be minimized for edge type r

$$O^r = KL(W^{(r)}_{uv}, s_t(u,v)) = -\sum_{(u,v)\in\mathcal{P}^r} W^{(r)}_{uv} \log s_r(u,v) + const, \tag{2}$$

where $KL(\cdot)$ stands for the Kullback-Leibler divergence.

Further, substituting Eq. (1) into Eq. (2) and taking all edge types into account and, the overall objective function becomes

$$O = -\sum_{\substack{(u,v)\in\mathcal{P}^r \\ r\in\mathcal{R}}} W^{(r)}_{uv} \log \frac{\exp(\boldsymbol{\mu}_r^\top \mathbf{g}_{uv})}{\sum_{\tilde{v}\in\mathcal{P}^r_{u*}} \exp(\boldsymbol{\mu}_r^\top \mathbf{g}_{u\tilde{v}}) + \sum_{\tilde{u}\in\mathcal{P}^r_{*v}} \exp(\boldsymbol{\mu}_r^\top \mathbf{g}_{\tilde{u}v})}. \tag{3}$$

To formulate edge embeddings required by Eq. (3), we derive from the same embeddings of the associated nodes regardless of the involved edge type, so that we reach a unified model where the learning process involving multiple edge types can work together and mutually enhance each other if they embody compatible semantics. While there are many options to build edge embedding from node embedding, we expect our formulation not to be over-complicated, so that the overall model could be computationally efficient and thereby easy-to-use. Moreover, in order for HEER to handle general HINs, it must also be able to handle directed and undirected edges accordingly. Considering these requirements, we

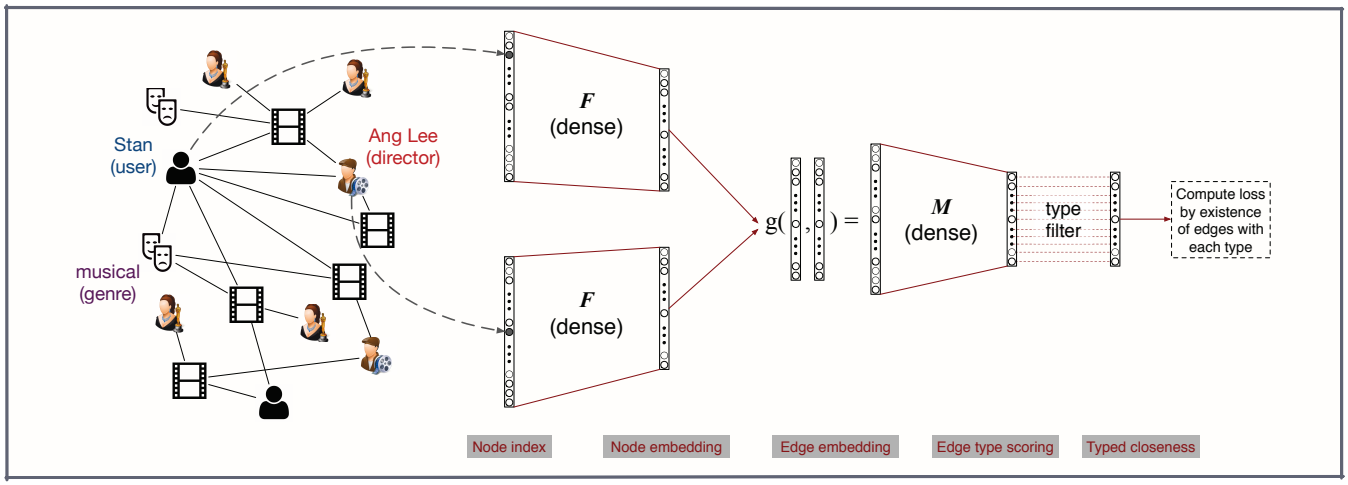


Figure 4: An illustration of the HEER architecture for learning HIN embedding via edge representation.

decompose node embedding into two sections $\mathbf{f}_u = \begin{bmatrix} \mathbf{f}_u^O \\ \mathbf{f}_u^I \end{bmatrix}$, where $\mathbf{f}_u^O$ and $\mathbf{f}_u^I$ are two column vectors of the same dimension, and build edge embedding on top of node embedding as

$$\mathbf{g}_{uv} \coloneqq \begin{cases} 2 \cdot \mathbf{f}_u^O \circ \mathbf{f}_v^I, & \text{directed representation from } u \text{ to } v, \\ \mathbf{f}_u^O \circ \mathbf{f}_v^O + \mathbf{f}_u^I \circ \mathbf{f}_v^I, & \text{undirected representation,} \end{cases} \tag{4}$$

where $\circ$ represents the Hadamard product. Besides Hadamard product, on can also build $\mathbf{g}_{uv}$ in a way similar to Eq. (4) using addition, subtraction, or outer-product. We leave the exploration of this direction to future works.

Taking Eq. (4) into account, learning node and edge embedding from an HIN by minimizing Eq. (3) is equivalent to the following optimization problem

$$\min_{\{\mathbf{f}_u\}_{u \in \mathcal{V}}, \{\boldsymbol{\mu}_r\}_{r \in \mathcal{R}}} O. \tag{5}$$

5.2 Model Inference

The HEER model in Eq. (5) that we aim to infer can be structured as a neural network as illustrated in Figure 4, where $\boldsymbol{F} = \left[\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{|\mathcal{V}|}\right] \in \mathbb{R}^{d_{\mathcal{V}} \times |\mathcal{V}|}$ and $\boldsymbol{M} = \left[\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_{|\mathcal{R}|}\right] \in \mathbb{R}^{d_{\mathcal{E}} \times |\mathcal{R}|}$. Each pair of nodes gets their respective embeddings through the dense layer $\boldsymbol{F}$, which further compose edge embedding by function $g(\cdot, \cdot)$. The raw scores for all edge types are obtained through another dense layer $\boldsymbol{M}$, followed by a type filter where the neuron for an edge type is connected to its corresponding neuron in the next layer only if this type is compatible with the node types of the input node pairs. Lastly, the loss is calculated by the typed closeness and the existence of edges in between the input node pair.

Since it is computationally expensive to compute the denominator in Eq. (1), we adopt the widely used negative sampling method [13], which enjoys linear-time computation. Specifically, each time, an edge between (u, v) with type r is sampled from the HIN with probability proportional to its weight. Then K negative node pairs $(u, \tilde{v}_i)$ and K negative node pairs $(\tilde{u}_i, v)$ are sampled, where each $\tilde{u}_i$ has the same type as u and each $\tilde{v}_i$ has the same type as v. The loss function computed from this sample becomes

$$\log \sigma(\boldsymbol{\mu}_r^\top \mathbf{g}_{uv}) + \sum_{i=1}^{K} \mathbb{E}_{\tilde{v}_i} \log \sigma(-\boldsymbol{\mu}_r^\top \mathbf{g}_{u\tilde{v}_i}) + \sum_{i=1}^{K} \mathbb{E}_{\tilde{u}_i} \log \sigma(-\boldsymbol{\mu}_r^\top \mathbf{g}_{\tilde{u}_i v}),$$

where $\sigma(\cdot)$ is the sigmoid function $\sigma(x) = \exp(x)/\big(1 + \exp(x)\big)$.

We adopt mini-batch gradient descent with the PyTorch implementation to minimize the loss function with negative sampling, where each mini-batch contains B sampled edges. We also use the node embeddings pretrained by the homogeneous network embedding algorithm LINE [29] to initialize the node embeddings in HEER. The edge-type–specific scoring vector $\boldsymbol{\mu}_r$ is initialized to be all-one vectors.

6 EXPERIMENTS

In this section, we evaluate the embedding quality of the proposed method and analyze the utility of employing edge representation and heterogeneous metric using two large real-world HINs. We first perform an edge reconstruction task to directly quantify how well the embedding algorithms can preserve the information in the input HINs. Then, we conduct in-depth case studies to analyze the characteristics of the proposed method.

6.1 Baselines

We compare the proposed HEER algorithm with baseline methods that fit the setting of our problem, *i.e.*, the methods should be applicable to general HINs without the help of additional supervision or expertise.

- **Pretrained** (LINE [29]). This baseline uses the LINE algorithm to generate node embeddings, which are also used to initialize HEER. LINE is a homogeneous network embedding algorithm based on the skip-gram model [13]. We use inner product to compute the score of observing an edge between a pair of node embeddings following the original paper [29].
- **AspEm** [23]. AspEm is a heterogeneous network embedding method that captures the incompatibility in HINs by decomposing the input HIN into multiple aspects with an unsupervised

measure using dataset-wide statistics. Embeddings are further learned independently for each aspect. This method considers the incompatibility in HINs but does not model different extent of incompatibility. Furthermore, it does not allow joint learning of embeddings across different aspects. Out of fairness, we let the number of aspects in AspEm to be two, in order to generate the final embedding with dimension that is identical to other methods. Inner product is also used to compute the score for this baseline.

- **UniMetrics** (metapath2vec++ [5]). This is a partial model of HEER, where the metrics $\{\boldsymbol{\mu}_r\}_{r\in\mathcal{R}}$ are not updated in the training process, *i.e.*, they remain uniform as initialized. It is equivalent to the metapath2vec++ [5] using all edges as length-1 meta-paths without further selection. This method restricts the negative sampling to be done within the consistent node types, *i.e.*, performs heterogeneous negative sampling, but still embeds all nodes into the same metric space regardless of types.
- **Pretrained + Logit**. On top of the embeddings from the previous Pretrained model, we train a logistic regression (Logit) model for each edge type using the input network. Then, we compute scores for test instances of each edge type using the corresponding Logit model. This method models heterogeneous metrics but does not allow the node embeddings and the edge embeddings to be further improved according to the inferred metrics.

6.2 Data Description and Experiment Setups

In this section, we describe the two real-world HINs used in our experiments as well as experiment setups.

Datasets. We use two publicly available real-world HIN datasets: DBLP and YAGO.

- **DBLP** is a bibliographical network in the computer science domain [30]. There are five types of nodes in the network: author, paper, key term, venue, and year. The key terms are extracted and released by Chen et al. [3]. The edge types include authorship (aut.), term usage (term), publishing venue (ven.), and publishing year (year) of a paper, and the reference relationship from a paper to another (ref.). We consider the first four edge types as undirected, and the last one as directed. The corresponding network schema is depicted in Figure 3b on page 4.
- **YAGO** is a large-scale knowledge graph derived from Wikipedia, WordNet, and GeoNames [24]. There are seven types of nodes in the network: person, location, organization, piece of work, prize, position, and event. A total of 24 edge types exist in the network, with five being directed and others being undirected. These edge types are illustrated together with the schema of the network in Figure 5.

We summarize the statistics of the datasets including the total number of nodes, the total number of edges, and the counts of each node type in Table 1.

Experiment Setups. For all experiments and all methods, we set the total embedding dimension to be 256. That is, for HEER and its related baselines, $\mathbf{f}_u \in \mathbb{R}^{256}$ and $\mathbf{f}_u^I, \mathbf{f}_u^O \in \mathbb{R}^{128}$, and each of the two

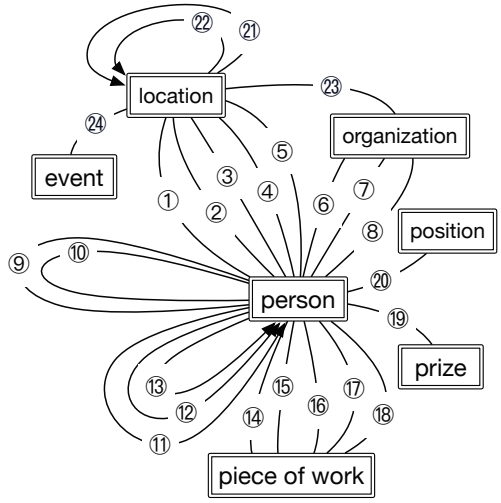


Figure 5: The schema of the YAGO network.

Table 1: Basic statistics for the DBLP and YAGO networks.

Dataset	Node	Edge	Node type	Edge type
DBLP	3,170,793	27,126,718	5	5
YAGO	579,721	2,191,464	7	24

aspects in AspEm uses a 128-dim embedding space. The pretrained model is always tuned to the best according to the performance in the edge reconstruction task to be introduced in Section 6.3. The negative sampling rate is always set to $K = 5$ for all applicable models. We always rescale the pretrained embedding by a constant factor of 0.1 before feeding them into HEER to improve the learning of heterogeneous metrics, which shares intuition with a previous study [14] in improving angular layout at the early stage of model training. The learning rate for gradient descent for HEER is set to 10 on both datasets. Note that we use the same set of hyperparameters for HEER on both DBLP and YAGO in order to provide an easy-to-use solution to the problem of comprehensive transcription of HINs without the hassle of extensive parameter tuning.

6.3 Edge Reconstruction Experiment

In order to directly quantify the extent to which an embedding algorithm can preserve the information in the input HINs, we devise the edge reconstruction experiments for both datasets. For each HIN, we first knock out a portion of edges uniformly at random, with a certain knock-out rate $\kappa \in (0, 1)$. Embedding of the network after knock-out is then learned using each compared method. The

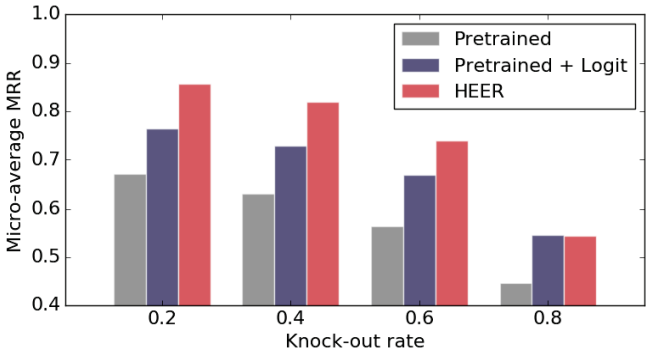
Table 2: Per-edge-type, micro-average, and macro-average MRR achieved by each model in the edge reconstruction task.

Dataset	DBLP							YAGO	
Metric (MRR)	Aut.	Term	Ref.	Pub. venue	Pub. year	Micro-avg.	Macro-avg.	Micro-avg.	Macro-avg.
Pretrained (LINE [29])	0.7053	0.4830	0.8729	0.7488	0.4986	0.6307	0.6617	0.7454	0.6890
AspEm [23]	0.7068	0.6010	0.8648	0.7612	0.6791	0.6976	0.7225	0.7832	0.6825
UniMetrics (len-1 metapath2vec++ [5])	0.7040	0.5772	0.8466	0.7534	0.6781	0.6812	0.7119	0.7437	0.6884
Pretrained + Logit	0.8187	0.6996	0.8072	0.8379	0.4889	0.7310	0.7304	0.8233	0.7012
HEER	**0.8964**	**0.7188**	**0.9573**	**0.9132**	**0.7421**	**0.8189**	**0.8456**	**0.8635**	**0.7185**

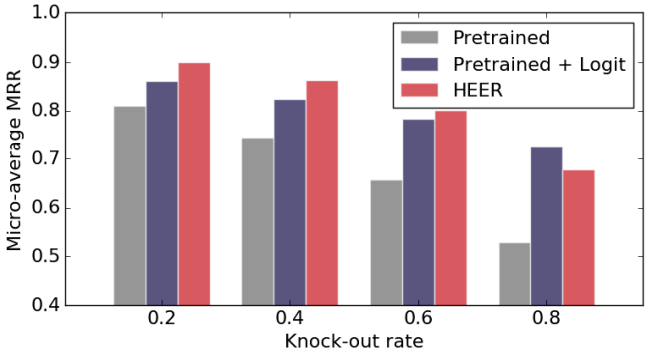
task is to reconstruct the edges being knocked out using the learned embedding models.

Specifically, for each edge that is knocked out from the network, suppose it is between node pair (u, v) and of edge type r, we randomly sample 10 negative pairs $(u, \tilde{v})$ that do not have type-r edges in the original full network, where $\tilde{v}$ is of the same node type as v. For any model after training, a score can be calculated to reflect the likelihood for each of the 11 node pairs to be associated by type-r edge in the current model. The reciprocal rank is then computed to measure the quality of the model, where the reciprocal rank is the reciprocal of the rank of the positive pair among all 11 node. Similarly, another reciprocal rank is computed for the same node pair (u, v) and 10 other randomly sampled negative pairs $(\tilde{u}, v)$ with fixed v but sampled $\tilde{u}$. Finally, we report the mean reciprocal rank (MRR), which is computed by the mean of reciprocal ranks for the target test instances. In particular, the micro-average MRR and the macro-average MRR are reported for both DBLP and YAGO, where the micro-average MRR is computed by the mean of all reciprocal ranks computed regardless of edge types, while the macro-average MRR is derived by first computing the mean of reciprocal ranks for each edge type, and then averaging all these means across different edge types. Additionally, we also report the MRR for each edge type for DBLP, since DBLP involves only 5 edge types, while YAGO has as many as 24 edge types. We present the results with knock-out rate $\kappa = 0.4$ in Table 2.

Modeling incompatibility benefits embedding quality. As shown in Table 2, the proposed HEER model outperformed all baselines in both datasets under both micro-average MRR and macro-average MRR, which demonstrated the effectiveness of the proposed method. Even when looking at each edge type in DBLP, the MRR achieved by HEER was still the best. Besides, in DBLP, AspEm outperformed Pretrained and UniMetrics on most metrics. Recall that AspEm decomposed the HIN into distinct aspects using dataset-wide statistics. As a result, it forbade semantically incompatible edge types to negatively affect each other in the embedding learning process and thereby achieved better results. In YAGO, the baselines considering heterogeneity did not always clearly outperform the simplest baseline Pretrained. We interpret this result by that YAGO has much more edge types than DBLP, which introduces even more varied extent of incompatibility, and the relatively simple approaches adopted by AspEm and UniMetrics in modeling incompatibility may not be enough to bring in significant performance boost. In contrast, armed with heterogeneous metrics fine-grained to the edge type level, HEER outperformed Pretrained by a clear margin even in YAGO.



(a) DBLP.



(b) YAGO.

Figure 6: Micro-average MRR under multiple knock-out rate κ in the egde reconstruction tasks.

Heterogeneous metrics helps improving embedding quality. As a sanity check, the Pretrained + Logit model helps rule out the possibility that HEER archives better results only by learning edge-type–specific metrics without actually improving embedding quality. From Table 2, it can be observed that by coupling with the additional edge-type–specific logistic regression and modeling heterogeneous metrics, the performance was improved on top of the Pretrained mode. This observation further consolidated the necessity of employing heterogeneous metrics for different edge types in solving the problem of comprehensive transcription of heterogeneous information networks. However, Pretrained + Logit still performed worse than the proposed HEER mode, which implies that the inferred heterogeneous metrics of HEER indeed in return improved the quality of the node and edge embedding.

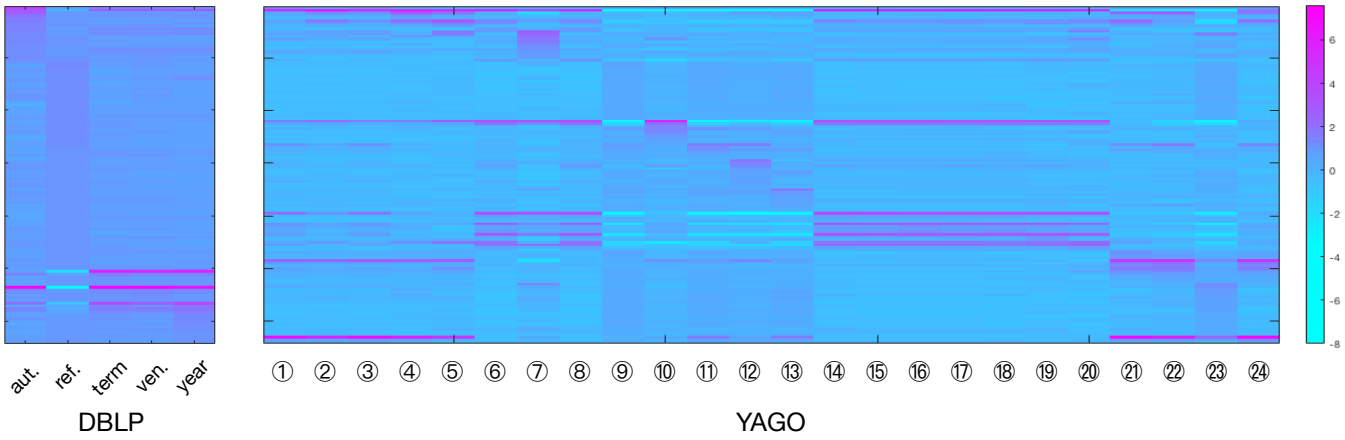


Figure 7: The learned heterogeneous metrics of the HEER model sensed the heterogeneous semantics of the HINs.

Heterogeneous negative sampling is not always enough to capture incompatibility. UniMetric performed better than the Pretained model in DBLP, it failed to make an absolute win over Pretained as other methods did in the YAGO dataset. Our interpretation of this result is that while the heterogeneous negative sampling as used by UniMetric does leverage certain type-specific information in HINs, it may not be always enough to model incompatibility and resolve the negative impact it brings to embedding quality. This observation is to be further corroborated in Section 6.4 by examining the capability of transcribing information implied by meta-paths in each model.

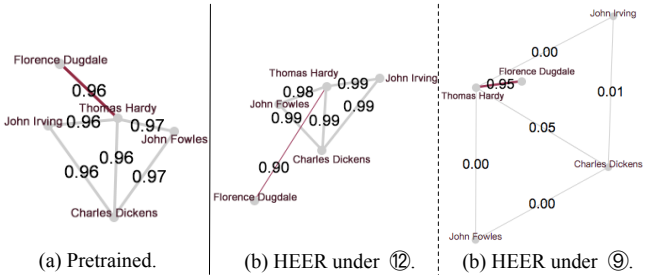
Varying Knock-Out Rate in Edge Reconstruction. Additionally, we vary the knock-out rate κ on both datasets and report the micro-average MRR for the proposed HEER model and two baseline models that require less training time. As presented in Figure 6, HEER outperformed all baselines under at most knock-out rates, which demonstrated the robustness of the proposed model. Besides, Pretrained + Logit outperformed Pretrained at all knock-out rates, which is also in line with the previous results we have presented. Notably, HEER did not outperform Pretrained + Logit when the knock-out rate $\kappa = 0.8$. This is explainable because only a very small portion (20%) of the original HIN was used for learning embedding when $\kappa = 0.8$. With a bigger model size than Pretrained + Logit, HEER was more prone to suffering from over-fitting.

6.4 Case Studies

In this section, we conduct a series of in-depth case studies to understand the characteristics of the proposed HEER model.

Learned heterogeneous metrics. HEER leverages heterogeneous metrics to model the different extent of incompatible semantics carried by different edge types. In this section, we analyzed the learned metrics $\{\boldsymbol{\mu}_r\}_{r\in\mathcal{R}}$ in HEER to verify if they indeed captured the different semantics and thereby enriched the model capability.

To this end, we use heat maps to illustrate $\{\boldsymbol{\mu}_r\}_{r\in\mathcal{R}}$ that are learned in the edge reconstruction experiments on both HINs. Specifically, for each dataset, we first standardize the elements of each $\boldsymbol{\mu}_r$ to have zero mean and unit deviation, so that $\boldsymbol{\mu}_r$'s have comparable scales after standardization for all $r \in \mathcal{R}$. Then, we re-order the $d_{\mathcal{E}}$ dimensions for better visualization and plot the heat maps in Figure 7.

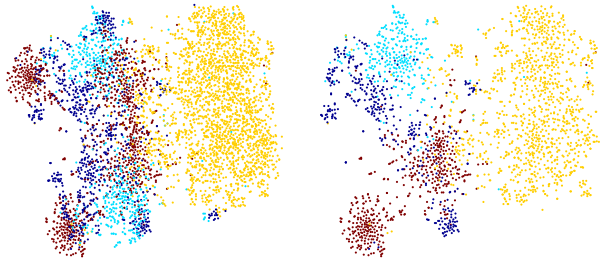


(a) Pretrained. (b) HEER under ⑫. (b) HEER under ⑨.

Figure 8: The subnetwork surrounding Thomas Hardy in YAGO in multiple embedding models under potentially different metrics.

Recall that the inferred metrics $\{\boldsymbol{\mu}_r\}_{r\in\mathcal{R}}$ were set to be all-one vectors in initialization, whereas Figure 7 shows that different metrics have generally reached different distributions over the $d_{\mathcal{E}}$ dimensions after training. This implies that the inferred metrics of the HEER model indeed sensed the heterogeneity of the HINs, and were using different projected metric spaces to embed different semantics of the input network. Notably, it can be seen from the heat map of YAGO that edge types ⑥ (isAffiliatedTo) and ⑧ (playsFor) have similar inferred metrics. This is actually expected because these edge types are often associated with the relationship between professional sports players and their associated teams in YAGO. Besides, similar phenomenon can be observed between ⑨ (isMarriedTo) and edge type ⑪ (hasChild).

Embedded subnetwork with different edge types. In order to understand how the network is impacted by the introduction of heterogeneous metrics, we took a closer look at a subnetwork surrounding the British writer Thomas Hardy in the YAGO dataset. Multiple other writers having the *influences* relationship (⑫, colored gray) with Thomas Hardy include Charles Dickens, John Fowles, and John Irving, while Florence Dugdale and Thomas Hardy enjoyed the *isMarriedTo* relationship (⑨, colored red). Besides, Fowles and



(a) UniMetrics. (b) HEER.

Figure 9: Visualization using t-SNE of 2,370 paper nodes that are linked to a given paper via four different meta-paths in DBLP. Each node is colored according to the meta-path it uses to reach the given paper. In Figure 9b, the nodes with same meta-path are clustered together, which implies HEER can preserve the information carried by different meta-paths in embedding learning even without the use of any meta-path. As a comparison, UniMetrics (len-1 metapath2vec++) yields less distinct clusters, with red, cyan, and blue nodes mingled together, showing adopting heterogeneous negative sampling without learned heterogeneous metrics is not sufficient to preserve the heterogeneity in the HIN embedding process.

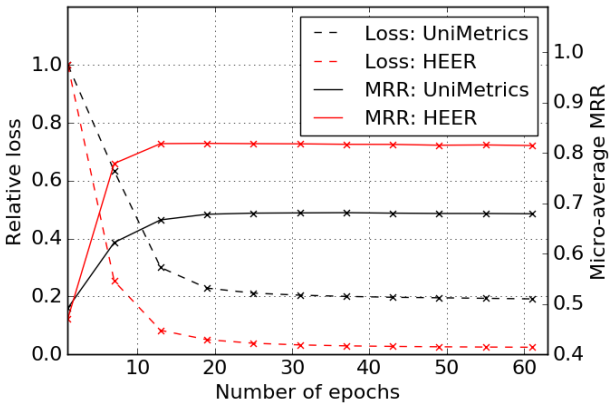


Figure 10: The relative loss and the micro-average MRR against the number of epochs for the proposed HEER model and the UniMetrics (len-1 metapath2vec) model.

Irving are also influenced by Dickens. In Figure 8, we visualized this subnetwork under each embedding model with the inferred possibility of edge existence marked, where the embedding models are training using the entire network. It can be seen from Figure 8a that without distinguishing edge types, Pretrained assigned high possibilities to all edges. Meanwhile, with the learned heterogeneous metrics, the HEER model assigned a relatively low probability for Dugdale and Hardy under the metric for *influences* as in Figure 8b (note that Dugdale was also a writer), and a clearly higher probability under the metric for *isMarriedTo* as in Figure 8c.

Transcription of information implied by meta-paths. When embedding each of the HINs, no meta-paths of length greater than 1 (*i.e.*., besides edges) were used, where a meta-path is the concatenation of certain edge types used to reflect certain semantics of an HIN [21, 25, 26]. We would like to verify whether the input HIN can be transcribed so comprehensively that the signals implied by meta-paths are also preserved in the embedding process, even without the use of guidance from meta-paths.

To this end, we visualize the embedding results considering four different meta-paths: [paper] $\overset{\text{ven.}}{\text{—}}$ [venue] $\overset{\text{ven.}}{\text{—}}$ [paper] (colored red), [paper] $\xrightarrow{\text{ref.}}$ [paper] (colored cyan), [paper] $\overset{\text{aut.}}{\text{—}}$ [author] $\overset{\text{aut.}}{\text{—}}$ [paper] (colored blue), and [paper] $\overset{\text{year}}{\text{—}}$ [year] $\overset{\text{year}}{\text{—}}$ [paper] (colored yellow). In particular, we consider a given paper node and find all paper nodes connected to this given paper by the aforementioned four meta-paths. Then we visualize the embeddings of all the nodes found in this way using the popular t-Distributed Stochastic Neighbor Embedding (t-SNE) [12] algorithm. Each node is colored according to the meta-path it uses to connect to the given paper node. Additionally, we randomly downsampled the group of nodes reached by meta-path [paper] $\overset{\text{year}}{\text{—}}$ [year] $\overset{\text{year}}{\text{—}}$ [paper] because a year can have edges with tens of thousands of paper nodes. The visualization results are shown in Figure 9.

In Figure 9b, the nodes with the same color are generally clustered together, which implies HEER can preserve the information implied by different meta-paths in embedding learning even without the use of any meta-path. As a comparison, we also visualized the same set of nodes in Figure 9a using embedding generated by UniMetrics (len-1 metapath2vec++), which yields less distinct clusters, with red, cyan, and blue nodes mingled together. Recall that UniMetrics also considers edge types when conducting negative sampling, and is different from HEER only in that the former does not employ heterogeneous metrics leaning. This result again demonstrated that adopting heterogeneous negative sampling without learned heterogeneous metrics is not sufficient to preserve the heterogeneity in the HIN embedding process, and is therefore not ideal for solving the problem of comprehensive transcription of heterogeneous information networks.

6.5 Efficiency Study

For efficiency study, we plot out the loss and the performance of the proposed HEER algorithm against the number of epochs in the edge reconstruction experiment. We also illustrate the same curves of the UniMetrics (len-1 metapath2vec++) model for comparison. The results are presented in Figure 10.

Judging from the curve for the loss again the number of epochs in Figure 10, HEER converges at a comparable rate with the skip-gram–based UniMetrics (metapath2vec++). Besides, HEER took less than twice as much time to finish each epoch as metapath2vec++ did. This is expected because HEER only additionally requires one-step gradient descent for one $\boldsymbol{\mu_r}$ when training on each sampled training example. As a result, the time complexity of HEER for each epoch differs from that of metapath2vec++ by a small constant factor. Combining the above two properties, HEER enjoys overall complexity linear to the number of nodes as skip-gram–based algorithms do [1, 5, 7].

7 CONCLUSION AND FUTURE WORKS

We studied the problem of the comprehensive transcription of HINs in embedding learning, which preserves the rich information in HINs and provides an easy-to-use approach to unleash the power of HINs. To tackle this problem, we identify that different extents of semantic incompatibility exist in real-world HINs, which pose challenges to the comprehensive transcription of HINs. To cope with these challenges, we propose an algorithm, HEER, that leverages edge representations and heterogeneous metrics. Experiments and in-depth case studies with large real-world datasets demonstrate the effectiveness of HEER and the utility of edge representations and heterogeneous metrics.

With the availability of edge representations proposed this paper, future works include exploration of more loss functions over edge representation, such as regression to model edges associated with ratings on HINs that have user-item reviews, or soft-max to model HINs where at most one edge type can exist between a pair of node types. One may also explore alternate ways to build edge embedding $\mathbf{g}_{uv}$ using addition, subtraction, outer-product, or deeper architectures. We leave the exploration of this direction to future works. Besides, it is also worthy of studying further boost the performance of HEER by incorporating higher-order structures such as network motifs, while retaining the advantage of HEER for being able to preserve the rich semantics from HINs.

Acknowledgments. This work was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). Any opinions, findings, and conclusions or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] Sami Abu-El-Haija, Bryan Perozzi, and Rami Al-Rfou. 2017. Learning Edge Representations via Low-Rank Asymmetric Projections. In *CIKM*.
[2] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architectures. In *KDD*.
[3] Ting Chen and Yizhou Sun. 2017. Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. In *WSDM*.
[4] Hanjun Dai, Bo Dai, and Le Song. 2016. Discriminative embeddings of latent variable models for structured data. In *ICML*.
[5] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.
[6] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. In *CIKM*.
[7] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*.
[8] Huan Gui, Jialu Liu, Fangbo Tao, Meng Jiang, Brandon Norick, and Jiawei Han. 2016. Large-scale embedding learning in heterogeneous event data. In *ICDM*.
[9] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. *arXiv preprint arXiv:1709.05584* (2017).
[10] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.
[11] Zemin Liu, Vincent W Zheng, Zhou Zhao, Fanwei Zhu, Kevin Chen-Chuan Chang, Minghui Wu, and Jing Ying. 2017. Semantic Proximity Search on Heterogeneous Graph by Proximity Embedding. In *AAAI*.
[12] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* 9, Nov (2008), 2579–2605.
[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.
[14] Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*. 6341–6350.
[15] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *ICML*.
[16] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric Transitivity Preserving Graph Embedding. In *KDD*. 1105–1114.
[17] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. 2016. Tri-party deep network representation. In *IJCAI*.
[18] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.
[19] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *KDD*.
[20] Jingbo Shang, Meng Qu, Jialu Liu, Lance M Kaplan, Jiawei Han, and Jian Peng. 2016. Meta-Path Guided Embedding for Similarity Search in Large-Scale Heterogeneous Information Networks. *arXiv preprint arXiv:1610.09769* (2016).
[21] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2017. A survey of heterogeneous information network analysis. *TKDE* 29, 1 (2017), 17–37.
[22] Yu Shi, Po-Wei Chan, Honglei Zhuang, Huan Gui, and Jiawei Han. 2017. PReP: Path-Based Relevance from a Probabilistic Perspective in Heterogeneous Information Networks. In *KDD*.
[23] Yu Shi, Huan Gui, Qi Zhu, Lance Kaplan, and Jiawei Han. 2018. AspEm: Embedding Learning by Aspects in Heterogeneous Information Networks.. In *SDM*.
[24] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.
[25] Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explorations* 14, 2 (2013), 20–28.
[26] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*.
[27] Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*.
[28] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*. ACM.
[29] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*.
[30] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *KDD*.
[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[32] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*.
[33] Linchuan Xu, Xiaokai Wei, Jiannong Cao, and Philip S Yu. 2018. On Exploring Semantic Meanings of Links for Embedding Social Networks. In *WWW*.
[34] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*. ACM.
[35] Baichuan Zhang and Mohammad Al Hasan. 2017. Name Disambiguation in Anonymized Graphs using Network Embedding. In *CIKM*.
[36] Chao Zhang, Liyuan Liu, Dongming Lei, Quan Yuan, Honglei Zhuang, Tim Hanratty, and Jiawei Han. 2017. TrioVecEvent: Embedding-Based Online Local Event Detection in Geo-Tagged Tweet Streams. In *KDD*. ACM.
[37] Muhan Zhang and Yixin Chen. Weisfeiler-Lehman neural machine for link prediction. In *KDD*. 575–583.
[38] Honglei Zhuang, Jing Zhang, George Brova, Jie Tang, Hasan Cam, Xifeng Yan, and Jiawei Han. 2014. Mining query-based subnetwork outliers in heterogeneous information networks. In *ICDM*.