

Heterogeneous Network Representation Learning

Yuxiao Dong¹, Ziniu Hu², Kuansan Wang¹, Yizhou Sun² and Jie Tang³

¹Microsoft Research, Redmond

²University of California, Los Angeles

³Tsinghua University, Beijing

{yuxdong, kuansanw}@microsoft.com, {bull, yzsun}@cs.ucla.edu, jietang@tsinghua.edu.cn

Abstract

Representation learning has offered a revolutionary learning paradigm for various AI domains. In this survey, we examine and review the problem of representation learning with the focus on heterogeneous networks, which consists of different types of vertices and relations. The goal of this problem is to automatically project objects, most commonly, vertices, in an input heterogeneous network into a latent embedding space such that both the structural and relational properties of the network can be encoded and preserved. The embeddings (representations) can be then used as the features to machine learning algorithms for addressing corresponding network tasks. To learn expressive embeddings, current research developments can fall into two major categories: shallow embedding learning and graph neural networks. After a thorough review of the existing literature, we identify several critical challenges that remain unaddressed and discuss future directions. Finally, we build the Heterogeneous Graph Benchmark¹ to facilitate open research for this rapidly-developing topic.

1 Introduction

The complex systems in real world are commonly associated with multiple types of objects and relations. Take the science enterprise as an example, it involves five types of entities, including papers, authors, institutes, venues, and research topics/fields. Other examples can be also found in social, biological, and economic systems. Usually, these complex systems can be effectively abstracted as heterogeneous networks, in which different types of vertices are connected under unique relations [Sun and Han, 2012]. As such, the science enterprise can be modeled as a heterogeneous academic network, such as the Open Academic Graph (OAG) [Zhang *et al.*, 2019b; Wang *et al.*, 2020; Tang *et al.*, 2008].

Formally, a heterogeneous network is defined as a directed graph $G = (V, E)$ in which each vertex $v \in V$ and each edge $e \in E$ are associated with their type mapping functions $\tau(v) : V \rightarrow T_V$ and $\phi(e) : E \rightarrow T_E$, respectively. T_V and T_E

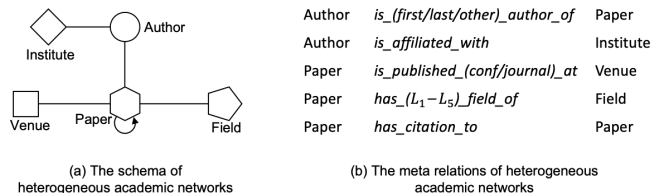


Figure 1: The schema and meta relations of Open Academic Graph (OAG)—reprinted from Figure 1 of [Hu *et al.*, 2020b].

represent the sets of vertex and edge types, satisfying $|T_V| + |T_E| > 2$. If two edges have the same relation type, they share the same vertex types for both their source vertices and target vertices. If both $|T_V|=1$ and $|T_E|=1$, it is a homogeneous network with the same types of vertices and edges.

The conventional mining process of heterogeneous networks usually starts with extracting typed structural features, which are further fed into machine learning models for downstream graph tasks. Commonly, this mining and learning workflow involves the unique concepts of meta relation, network schema, and meta path in heterogeneous networks, which are introduced as follows [Sun and Han, 2012].

Meta Relation. Each edge in a heterogeneous network is associated with its meta relation. For an edge $e = (s, t)$ linked from source vertex s to target vertex t , its meta relation is denoted as $\langle \tau(s), \phi(e), \tau(t) \rangle$.

Network Schema. Given a heterogeneous network G , its network schema $T_G = (T_V, T_E)$ consists of all meta relations in G , which are organized as a directed graph with G 's vertex types T_V as its vertex set and edge types T_E as its edge set.

Meta Path. A meta path scheme \mathcal{P} of G is defined as a sequence of meta relations over G 's network schema T_G . For example, an illustrative meta path in OAG is “author-paper-venue-paper-author”, which indicates the semantic of two authors publishing in the same venue.

For example, Figure 1 illustrates the network schema and meta relations of OAG. Based on these concepts and tools, various heterogeneous network mining methods have been proposed and applied for various graph-related applications, such as vertex classification, online recommendation, link prediction, community detection, and anomaly detection. A detailed review on this front is summarized in a recent monograph [Sun and Han, 2012].

Recently, the heterogeneous network mining paradigm

¹<https://github.com/HeterogeneousGraph>

has been shifted towards representation learning based techniques. The goal of heterogeneous network representation learning is to automatically learn latent low-dimensional embeddings for network objects, such as vertices, edges, and subgraphs. Its premise is that the intrinsic structural and semantic properties of the input network can be encoded into the latent embedding vectors and thus can benefit applications and tasks on heterogeneous networks. For example, the network embedding vectors of users and items can be used as the feature input of online recommendation systems.

In this survey, we review the recent progress in heterogeneous network representation learning and show how they have been shaping the mining and learning paradigm of heterogeneous network research. Specifically, we point out the challenges of the representation learning problem and then categorize the existing solutions into two groups, i.e., shallow embedding and graph neural networks (GNNs) based developments. Importantly, we identify the challenges and issues that remain unsolved. Finally, to encourage reproducible research on this topic, we build and release the Heterogeneous Graph Benchmark datasets at <https://github.com/HeterogeneousGraph>.

2 Heterogeneous Network Mining

We briefly introduce the history of heterogeneous network mining. Over the course of network science and graph mining development, there have been studies exploring networks that contain different types of vertices and edges, such as bipartite graphs in recommender systems [Koren *et al.*, 2009], hypergraphs [Zhou *et al.*, 2007], multi-layer networks [Kivelä *et al.*, 2014], and topic modeling in bibliographic networks [Tang *et al.*, 2008].

The formalization of heterogeneous networks can date back to the study of heterogeneous information networks [Sun *et al.*, 2009], in which the authors propose to generate clusters by utilizing links across different types of vertices. More importantly, the authors argue that “the interactions among multi-typed objects play a key role at disclosing the rich semantics that a network carries” [Sun and Han, 2012]. In addition to clustering, the importance of graph heterogeneity is also demonstrated in other graph mining tasks, including vertex classification, ranking [Ji *et al.*, 2011], similarity search [Sun *et al.*, 2011], link prediction [Dong *et al.*, 2015], anomaly detection [Chen *et al.*, 2016], etc.

The conventional workflow of mining heterogeneous networks is to firstly define meta paths or their variants, such as meta graphs or meta structures, over the network schema, and then use them as features to develop machine learning models [Sun and Han, 2012]. Take, for example, the link prediction task in academic networks, we can define the “author-paper-author” meta path to extract heterogeneous structural features and use them to infer whether there exist collaboration relationships between each pair of authors.

Challenges. The main challenge of heterogeneous network mining lies in the design of meta paths. Usually, to address a specific graph problem for an input network, we have to manually customize task- and data-specific meta paths, requiring related domain knowledge and experiences.

In addition, the conventional usage of meta paths is limited to the discrete space. If two vertices are not structurally connected in the graph, meta path based techniques cannot capture their relations. For example, suppose one scholar publishes papers all in NeurIPS and the other has all publications in ICML. According to the “author-paper-venue-paper-author” meta path, the similarity between them is zero, which is counter-intuitive due to the strong relevance between NeurIPS and ICML [Dong *et al.*, 2017]. This challenge resulted from discrete structures can be naturally addressed by using the latent representations in the continuous space.

3 Heterogeneous Network Representation

Over the past years, neural representation learning that aims to learn objects’ latent embeddings by using neural networks has offered revolutionized results for various domains, such as computer vision, natural language processing, and speech recognition. Inspired by this, there have been attempts to apply representation learning into networks. The premise of network representation learning is that instead of hand-crafting structural features, it can automatically embed the network structures into latent space, which can be then used for existing network mining tasks.

The main obstacle for neural representation learning on networks is the transformation of non-Euclidean graph structures into Euclidean embedding space, as there exists a gap between graph data and neural networks. Recently, the major progress to bridge this gap is to borrow ideas from graph theory. For example, one direction of attempts has been made to leverage random walks to transform graph structures into sequences, which can be consumed by sequence-based embedding learning algorithms, such as the DeepWalk and node2vec models [Perozzi *et al.*, 2014; Grover and Leskovec, 2016]. The other line of effort can be found on the connection between graph spectral and convolutional filters, with which we can design neural network operations directly over graphs, such as graph convolutional networks (GCN) [Kipf and Welling, 2017].

Most of representation learning techniques on heterogeneous networks are inspired by and developed based on (homogeneous) network representation learning. The core issue is how to transform the structures between different types of vertices and edges into latent spaces such that both the structural and semantic properties of heterogeneous networks can be encoded and preserved.

3.1 Heterogeneous Network Embedding

We review shallow learning-based embedding methods for heterogeneous networks. The “shallow” methods are characterized as an embedding lookup table, meaning that they directly encode each vertex as a vector, and this embedding table is the parameter to optimize. Along this line, methods can be categorized based on the assumptions they build upon.

Distributional Hypothesis based Methods. Skip-gram is a shallow embedding learning algorithm that is originally proposed to capture the semantic similarity of words in natural language [Mikolov *et al.*, 2013]. It is based on the distributional hypothesis [Harris, 1954], which states “words that

occur in the same contexts tend to have similar meanings”. Later on, the hypothesis is extended to networks: vertices that share the same structural contexts tend to be close in the embedding space. To capture it, several skip-gram based homogeneous network embedding methods have been proposed, such as DeepWalk and node2vec as well as their matrix factorization understanding NetMF [Qiu *et al.*, 2018].

To model the distributional hypothesis in heterogeneous networks, the PTE model [Tang *et al.*, 2015] first projects the input heterogeneous network into several homogeneous/bipartite networks. It then defines the vertex proximity in each of them with the assumption: “vertices with similar neighbors are similar to each other and thus should be represented closely in a low dimensional space.” Given this assumption, PTE applies the skip-gram framework on each network and optimizes all the embedding tables jointly.

To better capture the graph heterogeneity, the metapath2vec model is proposed [Dong *et al.*, 2017]. Its main idea is to amplify the random walk strategies, which are originally used to collect node contexts for modeling the distributional hypothesis in homogeneous networks. Specifically, it introduces the meta path-based random walk strategy, wherein the walker is restricted to transitions between particular types of vertices. Compared with PTE, this strategy can better capture the structural dependencies between different types of vertices in a unified way. For the embedding learning, metapath2vec directly leverages skip-gram with negative sampling to learn embeddings for all types of vertices in the heterogeneous networks. In addition, its enhanced version further presents type-dependent negative sampling for skip-gram, effectively modeling both the structural and semantic dependencies in heterogeneous networks.

Later on, various solutions leverage meta paths to construct heterogeneous contexts for learning representations for different downstream tasks. For example, similar to the phrase2vec extension of word2vec, HIN2vec [Fu *et al.*, 2017] directly considers meta paths as objects/contexts to learn embeddings for both vertices and meta paths. GATNE [Cen *et al.*, 2019] extends the framework to multiplex heterogeneous networks, and TapEm [Park *et al.*, 2019] further combines vertex pair embeddings with meta path embeddings. More recently, NeRank [Li *et al.*, 2019] is proposed to leverage the meta path-based skip-gram embedding model to help the question answering task with heterogeneous networks. HeteSpaceyWalk [He *et al.*, 2019] furthers meta path-based random walk by presenting the heterogeneous personalized spacey random walk algorithm with the guidance of meta paths, graphs, and schema. More importantly, this random walk strategy is associated with converged stationary distribution among vertices.

Other Methods. In addition to skip-gram based methods, other heterogeneous network embedding techniques include label propagation, factorization, and generative adversarial networks based methods. Moreover, most of these techniques are task-specific and require supervision information.

In 2014, one of the very early attempts to learn representations for heterogeneous networks is the label propagation-based LSHM model [Jacob *et al.*, 2014]. LSHM borrows

the smoothness assumption of label propagation in homogeneous networks—“two connected nodes tend to have the same labels”—for heterogeneous networks. Its idea is to enforce vertices (of different types) to have similar representations if they are connected by using the vertex-type specific classification loss. The method is under the label propagation framework for vertex classification. Further, a similar joint optimization of task and structure losses is presented to address the heterogeneous link prediction problem [Chen and Sun, 2017]. Its structure loss is constructed based on the meta paths that are manually designed for specific tasks in the heterogeneous networks. In summary, these methods are (semi-)supervised and the authors argue that the structural and label dependencies between different types can be modeled.

Similar to PTE, the HERec model also projects an input heterogeneous network into several dimensions according to customized meta paths [Shi *et al.*, 2017]. The vertex embeddings are then learned by using the metapath2vec framework within each projection, all of which are merged by a factorization based fusion model with supervised information from the recommendation task. Differently, the HueRec method [Wang *et al.*, 2019b] argues that there exist interrelations between different meta paths and thus it unifies them into an end-to-end framework for the recommendation task in heterogeneous networks.

Most skip-gram based embedding methods are associated with negative sampling for scalable and fast learning. More broadly, this technique lies in the domain of noise contrastive estimation (NCE), built on which the APE model [Chen *et al.*, 2016] efficiently learns the interaction probability between different vertex types with the “context-dependent” noise distribution regardless of large data space. Further, a more recent study proposes generative adversarial networks (GAN) based framework HeGAN for achieving this [Hu *et al.*, 2019]. Instead of direct sampling, the goal of HeGAN is to generate the underlying vertex distribution and thus to produce better negative samples with heterogeneous structures in mind.

Summary. In the context of shallow embedding, there are both unsupervised and supervised learning frameworks for heterogeneous networks. Most of the unsupervised methods are built upon the application of distributional hypothesis into heterogeneous networks, such as skip-gram based models. For (semi-)supervised frameworks, the learning objective is usually a combination of heterogeneous structure modeling and task-specific optimization.

Nevertheless, under both learning settings, the majority of methods rely on the manual exploration of heterogeneous structures, i.e., the selection of meta paths or variants, for capturing the structural and semantic dependencies.

3.2 Heterogeneous Graph Neural Networks

In the past few years, Graph Neural Networks (GNNs) have shown promising results for modeling structural and relational data. Different from shallow embedding methods, GNNs are often empowered by more complex encoders, usually a deep neural network, enabling the natural modeling of both structures and vertex attributes. In this part, we summarize the latest development of GNNs for heterogeneous network representation learning.

We first introduce the general GNN framework for relational data. One way is to view it as a neural message passing framework [Gilmer *et al.*, 2017], in which the structure of the input graph is used as the backbone for propagating (vertex or edge) information in order to form expressive representations. Or, we can consider it as the feature aggregation process [Bruna *et al.*, 2014; Hamilton *et al.*, 2017], wherein for each vertex, its neighborhood is considered as a receptive field from which it aggregates information recursively.

Formally, the aggregation process can be abstracted as $\mathbf{H}^{(l+1)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$, where $\mathbf{H}^{(l)}$ and $\mathbf{W}^{(l)}$ are the (hidden) node representation and parameter matrix of the l^{th} layer, $\sigma(\cdot)$ is usually a non-linear function, and $\hat{\mathbf{A}}$ represents a filter derived from some form of the graph structure. For example, in GCN, it is the symmetric normalized adjacency matrix [Kipf and Welling, 2017]. Given this framework, a line of research has been proposed to better learn network representations. Next, we review the recent developments on heterogeneous graph neural networks.

Relational GCN. One of the first attempts to model the multi-relational graph using GNN is the Relational Graph Convolutional Networks (RGCN) [Schlichtkrull *et al.*, 2018], which keeps a distinct linear projection weight for each edge type. To deal with graphs with a large number of relations, RGCN also decomposes the relation-specific parameters as a linear combination of several base matrices.

Another notable effort is the Decagon model [Zitnik *et al.*, 2018], which applies specific graph convolutional filters over each type of relations in a multimodal graph with proteins, drugs, polypharmacy side effects as vertices. Note that the relation-specific parameters in Decagon are shared for different vertices.

Heterogeneous GNNs. To deal with heterogeneous graph structures and node attributes, the Heterogeneous Graph Neural Network (HetGNN) [Zhang *et al.*, 2019a] uses type-specific RNNs to encode features for each type of neighbor vertices, followed by another RNN to aggregate the encoded neighbor representations of different types.

Heterogeneous Graph Attention. To incorporate attention [Bahdanau *et al.*, 2015; Veličković *et al.*, 2017] into the heterogeneous GNN design, the GEM method [Liu *et al.*, 2018] leverages distinct aggregators for each relation and learns the attention coefficients for modeling the importance of different types of vertices.

Later, the Heterogeneous Graph Attention Network (HAN) [Wang *et al.*, 2019a] uses meta paths as edges to augment the graph, and maintains different weight matrices for each meta-path-defined edge. It also utilizes the high-level semantic attention to differentiate and aggregate information from different meta paths.

Heterogeneous Graph Transformer. All the previous works use either node vertex or edge type alone to determine GNN weight matrices. However, for those relations that do not have sufficient occurrences, it is hard to learn accurate relation-specific weights. To address this, the Heterogeneous Graph Transformer (HGT) [Hu *et al.*, 2020b] uses each edge’s meta relation to parameterize the Transformer-like self-attention

architecture [Vaswani *et al.*, 2017], so that both the common and specific patterns of different relationships are captured by using equal or even fewer parameters. In addition, different from most of the existing works that require the manual design of meta paths, HGT can automatically learn the importance of implicit meta paths by the nature of multi-layer GNNs to incorporate high-order heterogeneous neighbor information.

Another concurrent work, the Graph Transformer Networks [Yun *et al.*, 2019], learns a soft selection of edge types and composite relations for generating useful multi-hop meta paths to augment the input graph. After that, it adopts the vanilla GCN layer to the augmented graph for getting contextualized vertex embeddings.

Summary. The nature of graph neural networks ensure all the aforementioned methods to get inductive embeddings for any unseen vertices or graphs, instead of maintaining a large lookup table for each vertex within the graph as shallow embedding methods. Also, they can naturally leverage rich vertex attributes by neural network encoders.

To adapt to the heterogeneous setting, all existing models adopt different weight matrices for each relational type. Most of them leverage human-designed meta paths to augment the graphs, while some recent works utilize the self-attention mechanism to automatically learn “soft” meta paths.

3.3 Knowledge Graphs & Attributed Networks

Heterogeneous networks are closely connected with knowledge graphs and attributed networks. In this short survey, we don’t dig deep into their differences and connections. Generally, knowledge graph embedding is a technique used to complete the knowledge graph triplets. Its major solutions can be categorized into two classes: neural embedding—e.g., TransE [Bordes *et al.*, 2013]—and tensor factorization—e.g., RESCAL [Nickel *et al.*, 2012]—based models. Attributed network embedding is given the similar input as graph neural networks. Notably, the ANE and DANE models [Li *et al.*, 2017] are proposed to model the inherent correlations between network structures and attributes, with the extension to dynamic environments. Recently, Cen *et al.* formalize the representation learning problem for attributed multiplex heterogeneous networks. To address the attributed, multiplex, and heterogeneous challenges, they present the GATNE framework with a theoretical demonstration of its expressiveness superiority [Cen *et al.*, 2019].

3.4 Applications

Since heterogeneous networks can abstract and model real-world complex systems, representation learning on heterogeneous networks have numerous applications, such as similarity search, knowledge graph reasoning, question answering and recommendation. For example, the metapath2vec method has been deployed for similarity search in Microsoft Academic². Another industry system that is benefited from the heterogeneous network embedding is the online recommendation module in e-commerce [Cen *et al.*, 2019].

² An illustrative example of the Related Journals function for *nature* can be found at <https://academic.microsoft.com/journal/137773608>.

Dataset	#nodes	#edges	#papers	#authors	#fields	#venues	#institutes	#P-A	#P-F	#P-V	#A-I	#P-P
NN	66,211	3,638,702	18,911	32,307	9,540	2,008	3,445	48,551	202,281	189,116	48,765	1,355,082
CS	11,732,027	107,263,811	5,597,605	5,985,759	119,537	27,433	16,931	15,571,614	47,462,559	5,597,606	7,190,480	31,441,552
OAG	178,663,927	2,236,196,802	89,606,257	88,364,081	615,228	53,073	25,288	300,853,688	657,049,405	89,606,258	167,449,933	1,021,237,518

Table 1: Open Academic Graph (OAG) Statistics.

4 Challenges, Directions, and Open Data

Over the course of its development, representation learning has empowered a paradigm shift for mining (heterogeneous) networks. It has been particularly evident that graph neural networks can significantly re-shape heterogeneous network representation learning and thus benefit all possible applications on heterogeneous network data. Therefore, we discuss remaining challenges and future directions with an emphasis on GNN based techniques.

4.1 Future Directions

Avoiding Meta Path Design. The broad premise of representation learning is that feature engineering which involves manual effort and domain expertise can be avoided. As we discussed above, most of heterogeneous representation learning methods require the manual customization of meta paths or their variants (e.g., meta graphs) for the models to capture the semantics underlying the given heterogeneous structures. For example, metapath2ve, HetGNN, and HAN have to take pre-defined meta paths as input to address the dedicated tasks. Essentially, their representation capacities are limited to specific tasks on specific data. One promising attempt to avoid meta path design is the recent HGT model [Hu *et al.*, 2020b], where the issue is naturally resolved by GNNs’ feature propagation across multiple layers. Therefore, to fully unleash the power of representation learning, it would be necessary to automatically learn heterogeneous network representations without meta path pre-design, that is, avoiding the pre-exploration of network structures and semantics.

Multi-Sense Representations. To date, most embeddings are learned for one single sense from the complex and multi-typed heterogeneous networks. Take, for example, the distributional hypothesis, there is no distinction between the structural contexts. Specifically, if we consider the scholar “Jiawei Han” as an illustration, we can name a few different senses for him. The most similar scholars to him under the sense of scientific collaboration are “Jian Pei” and “Xifeng Yan”, who are his Ph.D. students. Or, under the sense of venue publications, “Philip S. Yu” and “Christos Faloutsos” are most similar to him. Or, under the sense of scientific impact for data mining, he is more similar to “Rakesh Agrawal” in Database and “Bruce Croft” in IR. From this example, we can clearly observe that there exist multi-sense representations in heterogeneous networks. How to define, learn, and use them remains largely open and unexplored.

Pre-Training of (Heterogeneous) GNNs. As reviewed in previous section, existing (heterogeneous) GNNs are end-to-end trained in (semi-)supervised settings. In other words, we need the dedicated label data for each task on each dataset,

which is usually expensive to achieve in the real world, particularly for Web-scale heterogeneous graphs. To solve the label scarcity issue, pre-trained neural models have been adopted and significantly advanced the development of various AI domains, such as BERT [Devlin *et al.*, 2019] in natural language processing. The benefits of a pre-trained model include 1) most of the time, it is trained in an unsupervised (or self-supervised) manner, requiring no task- and data-specific label information; 2) it can be used for fine-tuning over different downstream tasks, even being generalizable to various data.

Herein, we envision two types of (heterogeneous) GNN pre-training directions for future studies. First, the pre-trained model is learned to capture the underlying structural properties of an input network such that it can be used for different tasks on the same input data. Second, we pre-trained a GNN model on a set of networks (of different types) intending to capture the intrinsic structural properties across these networks and thus facilitating downstream tasks on different (unseen) networks. Naturally, the pre-training of heterogeneous GNNs also faces the challenge of graph heterogeneity, making it different from and more challenging than existing pre-training techniques in homogeneous text/image/speech data.

Multi-Task Learning. Recently, multi-task learning has enabled deep learning methods to achieve superiority over traditional single-task learning for learning representations in NLP and CV. Its idea is to address multiple tasks in parallel to recognize and leverage the similarities/distinctions across these tasks. Research has reported that in doing so, the performance as a whole can be improved when compared to task-specific models [Ruder, 2017; Wu *et al.*, 2020].

Notice that given a heterogeneous network of multiple types of vertices and edges, it is usually associated with various mining tasks. For example, there are many real-world applications on a heterogeneous academic network, such as name disambiguation, the inference of papers’ topics or authors’ research interests, literature and citation recommendation, similarity search, and so on. All those heterogeneous network tasks could benefit from multi-task learning, in which a shared representation is learned across tasks and task-specific parameters can contribute to other tasks. The unique property of heterogeneous networks is that they contain multiple types of vertices and edges, which might be a natural application of multi-task learning.

Dynamics and Scalability. In the real world, most networks evolve over time, whose modeling has long been a research challenge. The network dynamics issue becomes particularly challenging when it is coupled with different types of vertices and edges—heterogeneous networks. Most existing methods handle it by splitting the input dynamic network into multi-layer networks based on timestamps. Consequently, the struc-

GNN Models			GCN	RGCN	GAT	HetGNN	HAN	HGT
NN	Paper-Field	NDCG	.486±.023	.491±.031	.519±.047	.511±.057	.526±.056	.548±.046
		MRR	.460±.038	.472±.036	.496±.042	.488±.047	.504±.052	.513±.045
	Paper-Venue	NDCG	.478±.053	.483±.062	.497±.070	.492±.066	.504±.068	.513±.062
		MRR	.352±.066	.371±.073	.379±.068	.374±.072	.386±.069	.397±.067
	Author Disambiguation	NDCG	.883±.021	.902±.028	.926±.032	.922±.028	.931±.036	.942±.035
		MRR	.807±.028	.828±.042	.845±.036	.839±.038	.848±.041	.857±.037
CS	Paper-Field	NDCG	.344±.021	.322±.053	.357±.058	.346±.071	.352±.051	.403±.041
		MRR	.353±.053	.340±.061	.382±.057	.373±.051	.388±.065	.439±.078
	Paper-Venue	NDCG	.406±.081	.412±.076	.437±.082	.431±.074	.449±.072	.473±.054
		MRR	.215±.066	.216±.105	.239±.089	.245±.069	.254±.074	.288±.088
	Author Disambiguation	NDCG	.826±.039	.835±.042	.864±.051	.850±.056	.859±.053	.894±.034
		MRR	.661±.045	.665±.054	.694±.052	.668±.061	.688±.049	.732±.038
OAG	Paper-Field	NDCG	.318±.034	.328±.046	.339±.049	.336±.062	.342±.051	.367±.048
		MRR	.322±.047	.332±.052	.348±.045	.350±.053	.358±.049	.378±.071
	Paper-Venue	NDCG	.302±.066	.313±.051	.317±.057	.309±.071	.327±.062	.355±.062
		MRR	.194±.070	.193±.047	.196±.052	.192±.059	.214±.067	.247±.061
	Author Disambiguation	NDCG	.738±.042	.755±.048	.797±.044	.803±.058	.821±.056	.852±.048
		MRR	.612±.064	.619±.057	.645±.063	.649±.052	.660±.049	.688±.054

Table 2: Experimental results on three benchmark datasets—partial results on OAG are reprinted from [Hu *et al.*, 2020b].

tural and temporal dependencies across different layers are usually ignored. To this end, how to effectively model dynamic heterogeneous networks is still an open question, not to mention representation learning for them.

Another common challenge lies in the large scale of real-world graphs. OAG consists of billions of vertices and edges, spanning between 1800 and 2020 (to date). Other Web-scale graphs are even more large, such as the LinkedIn Economic Graph, Wikipedia Entity Graph, Microsoft Office Graph, etc. Existing effort mainly focuses on improving the scalability of homogeneous network representation learning methods, such as NetSMF [Qiu *et al.*, 2019], ProNE [Zhang *et al.*, 2019c], and PinSage [Ying *et al.*, 2018]. For most existing heterogeneous network representation learning techniques, it is prohibitively expensive to learn over Web-scale graphs. Therefore, scalable and computational feasible models form a critical direction for future work.

Others. In addition to the aforementioned issues specific to heterogeneous networks, there are also challenges faced by general GNN models, such as the over-smoothing [Li *et al.*, 2018] and overfitting issues when many GNN layers are stacked. The resolve of these problems can naturally benefit heterogeneous network representation learning.

4.2 Open Data and Benchmark

Previously, we discussed the major technical challenges in heterogeneous network representation learning. Additionally, the community also faces the data challenge, though there are several public heterogeneous network datasets, such as the DBLP/AMiner data, Amazon/Yelp review data, and IMDB data. First, these datasets are used differently with different tasks and evaluation metrics. Second, they are pre-processed randomly, such as different versions, different training-testing-validation splits, different selections of vertex types, etc. Finally, even on the same data with the same pre-processing, different results are reported across literature.

To address these issues, we propose to build the Heterogeneous Graph Benchmark³ with the first release of three datasets built from OAG. The idea is similar to the Open Graph Benchmark [Hu *et al.*, 2020a] and GNN benchmark [Dwivedi *et al.*, 2020], but with the focus on heterogeneous graphs. The current datasets are constructed based on different (high-level) research fields in OAG, including Artificial Neural Networks (NN), Computer Science (CS), and all topics (OAG). The statistics of the three datasets are listed in Table 1. For each of them, we build three tasks with the pre-defined data splitting strategy and feature definition, including the prediction of a paper’s research field and publication venue, as well as author name disambiguation.

Moreover, we run several popular and powerful (heterogeneous) graph neural networks, including GCN, GAT, RGCN, HetGNN, HAN, and HGT, to report the statistically significant results for each task on each dataset in Table 2, which can be referred to as benchmark performance.

5 Conclusion

In this work, we introduce the problem of heterogeneous network representation learning and discuss its underlying challenges. We review the recent progress in addressing these challenges. More importantly, we dive deeper into the problem and unveil the critical issues and pain-points that should be resolved in the future. To facilitate open and reproducible research on this rapidly-developing topic, we take the first step to release the Heterogeneous Graph Benchmark for the research community.

Acknowledgements. Ziniu Hu and Yizhou Sun are supported by NSF III-1705169, NSF CAREER Award 1741634, NSF #1937599, Okawa Foundation Grant, and Amazon Research Award. Jie Tang is supported by the NSFC for Distinguished Young Scholar (61825602).

³<https://github.com/HeterogeneousGraph>

References

- [Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
- [Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.
- [Bruna *et al.*, 2014] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.
- [Cen *et al.*, 2019] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. In *KDD*, 2019.
- [Chen and Sun, 2017] Ting Chen and Yizhou Sun. Task-guided and path-augmented heterogeneous network embedding for author identification. In *WSDM*, 2017.
- [Chen *et al.*, 2016] Ting Chen, Lu-An Tang, Yizhou Sun, Zhengzhang Chen, and Kai Zhang. Entity embedding-based anomaly detection for heterogeneous categorical events. In *IJCAI*, 2016.
- [Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.
- [Dong *et al.*, 2015] Yuxiao Dong, Jing Zhang, Jie Tang, Nitesh V. Chawla, and Bai Wang. CoupledLp: Link prediction in coupled networks. In *KDD*, 2015.
- [Dong *et al.*, 2017] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. meta-path2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 2017.
- [Dwivedi *et al.*, 2020] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.
- [Fu *et al.*, 2017] Tao-Yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*, 2017.
- [Gilmer *et al.*, 2017] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017.
- [Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *KDD*, 2016.
- [Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.
- [Harris, 1954] Zellig S. Harris. Distributional structure. *WORD*, 1954.
- [He *et al.*, 2019] Yu He, Yangqiu Song, Jianxin Li, Cheng Ji, Jian Peng, and Hao Peng. Hetspacewalk: A heterogeneous spacey random walk for heterogeneous information network embedding. In *CIKM*, 2019.
- [Hu *et al.*, 2019] Binbin Hu, Yuan Fang, and Chuan Shi. Adversarial learning on heterogeneous information networks. In *KDD*, 2019.
- [Hu *et al.*, 2020a] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.
- [Hu *et al.*, 2020b] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *WWW*, 2020.
- [Jacob *et al.*, 2014] Yann Jacob, Ludovic Denoyer, and Patrick Gallinari. Learning latent representations of nodes for classifying in heterogeneous social networks. In *WSDM*, 2014.
- [Ji *et al.*, 2011] Ming Ji, Jiawei Han, and Marina Danilevsky. Ranking-based classification of heterogeneous information networks. In *KDD*, 2011.
- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [Kivelä *et al.*, 2014] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.
- [Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.
- [Li *et al.*, 2017] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. Attributed network embedding for learning in a dynamic environment. In *CIKM*, 2017.
- [Li *et al.*, 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018.
- [Li *et al.*, 2019] Zeyu Li, Jyun-Yu Jiang, Yizhou Sun, and Wei Wang. Personalized question routing via heterogeneous network embedding. In *AAAI*, 2019.
- [Liu *et al.*, 2018] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. Heterogeneous graph neural networks for malicious account detection. In *CIKM*, 2018.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.
- [Nickel *et al.*, 2012] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *WWW*, 2012.
- [Park *et al.*, 2019] Chanyoung Park, Donghyun Kim, Qi Zhu, Jiawei Han, and Hwanjo Yu. Task-guided pair embedding in heterogeneous network. In *CIKM*, 2019.
- [Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.
- [Qiu *et al.*, 2018] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*, 2018.
- [Qiu *et al.*, 2019] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. Netsmf: Large-scale network embedding as sparse matrix factorization. In *WWW*, 2019.
- [Ruder, 2017] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.
- [Schlichtkrull *et al.*, 2018] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, 2018.
- [Shi *et al.*, 2017] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. Heterogeneous information network embedding for recommendation. *IEEE TKDE*, 2017.
- [Sun and Han, 2012] Yizhou Sun and Jiawei Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers, 2012.
- [Sun *et al.*, 2009] Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *EDBT*, 2009.
- [Sun *et al.*, 2011] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, 2011.
- [Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, 2008.
- [Tang *et al.*, 2015] Jian Tang, Meng Qu, and Qiaozhu Mei. PTE: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, 2015.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [Wang *et al.*, 2019a] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *KDD*, 2019.
- [Wang *et al.*, 2019b] Zekai Wang, Hongzhi Liu, Yingpeng Du, Zhonghai Wu, and Xing Zhang. Unified embedding model over heterogeneous information network for personalized recommendation. In *IJCAI*, 2019.
- [Wang *et al.*, 2020] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020.
- [Wu *et al.*, 2020] Sen Wu, Hongyang R Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *ICLR*, 2020.
- [Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, 2018.
- [Yun *et al.*, 2019] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. Graph transformer networks. In *NeurIPS*, 2019.
- [Zhang *et al.*, 2019a] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *KDD*, 2019.
- [Zhang *et al.*, 2019b] Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, and Kuansan Wang. OAG: toward linking large-scale heterogeneous entity graphs. In *KDD*, 2019.
- [Zhang *et al.*, 2019c] Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. Prone: fast and scalable network representation learning. In *IJCAI*, 2019.
- [Zhou *et al.*, 2007] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *NeurIPS*, 2007.
- [Zitnik *et al.*, 2018] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 2018.