Representation Learning on Knowledge Graphs for Node Importance Estimation

Han Huang[1], Leilei Sun[1,2*], Bowen Du[1,2], Chuanren Liu[3], Weifeng Lv[1,2], Hui Xiong[4]
[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2]Peng Cheng Laboratory, Shenzhen 518055, China
[3]Department of Business Analytics and Statistics, University of Tennessee, Knoxville, TN 37996, USA
[4]Department of Management Science and Information Systems, Rutgers University, NJ 07102, USA
[1]{h-huang,leileisun,dubowen,lwf}@buaa.edu.cn, [3]cliu89@utk.edu, [4]hxiong@rutgers.edu

ABSTRACT

In knowledge graphs, there are usually different types of nodes, multiple heterogeneous relations, and numerous attributes of nodes and edges, which impose the challenges on the task of Node Importance Estimation (NIE). Indeed, existing NIE approaches, such as PageRank (PR) and Node-Degree (ND), are not designed for handling knowledge graphs with the rich information related with these multifarious nodes and edges. To this end, in this paper, we propose a representation learning framework to leverage the rich information inherent in these multifarious nodes and edges for improving node importance estimation in knowledge graphs. Specifically, we provide a **R**elational **G**raph **T**ransformer **N**etwork (RGTN), where a relational graph transformer is first proposed to propagate node information with the consideration of semantic predicate representations. Here, the assumption is that different predicates may have distinct effects on the transmission of node importance. Then, two separate encoders are designed to capture both the structural and semantic information of nodes respectively, and a co-attention module is developed to fuse the two separate representations of nodes. Next, an attention-based aggregation module is adopted to map the representations of nodes to their importance values. In addition, a learning-to-rank loss is designed to ensure that the learned representations can be aware of the relative ranking information among nodes. Finally, extensive experiments have been conducted on real-world knowledge graphs, and the results illustrate that our model outperforms the existing methods consistently for all the evaluation metrics. The code and the data are available at https://github.com/GRAPH-0/RGTN-NIE.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

Node Importance, Representation Learning, Knowledge Graphs, Neural Network

ACM Reference Format:
Han Huang, Leilei Sun, Bowen Du, Chuanren Liu, Weifeng Lv, Hui Xiong. 2021. Representation Learning on Knowledge Graphs for Node Importance Estimation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3447548.3467342

1 INTRODUCTION

Node Importance Estimation (NIE) is a task of inferring the significance or the popularity of a node in a graph according to the structure and attribute information. It is a crucial research topic in data mining that has been extensively studied with various applications such as web searching, recommender systems, opinion leader identification, and source allocation [3, 4, 8, 10, 14, 16–18, 27].

The existing methods for node importance estimation could be divided into two categories: hand-designed methods and machine-learned methods. In network science, many indicators have been proposed to evaluate the node importance. Take the centrality indicators [3, 4, 8] for example, which view a graph as a description of the paths where the information flow. The degree centrality counts the number of walks within one step, while the betweenness centrality counts the shortest paths which pass through the given node. PageRank (PR) is a successful method on website importance estimation [16]. With the underlying assumption that the more important nodes are likely to receive more links from other nodes, PR counts the number and the quality of edges linked to a node to determine a rough estimate of the node importance.

However, these methods could not be directly used to evaluate the importance of nodes in complex knowledge graphs because they only take the graph topology into account and ignore the abundant structure and semantic information contained in the multifarious nodes and edges. For an instance of knowledge graphs shown in Figure 1, besides the entities represented by nodes, there are also multi-relational edges and text descriptions related to the nodes. Take the person entity "Guy Ritchie" as an example. This entity has two links with the movie entity "The Gentlemen", indicating that "Guy Ritchie" "directed" and "wrote" "The Gentlemen". In addition, the biography for "Guy Ritchie" and the plot summary for "The Gentlemen" further enrich the node contents. In this scenario, the node degree or the importance value computed by PageRank could

∗ Corresponding Author.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467342

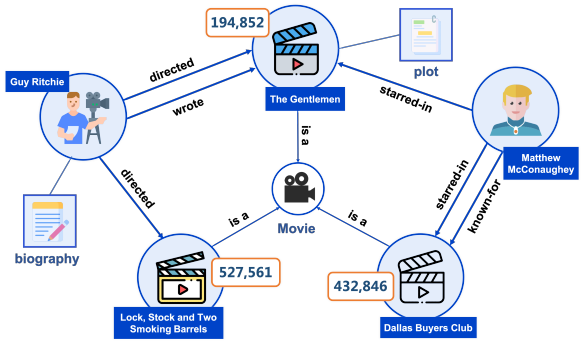

Figure 1: An example of movie knowledge graphs. It contains rich information including entities, predicates and node descriptions. The number in an orange box is the number of votes for the movie, reflecting the node importance. Some potential applications for NIE can refer to Section 2.2 and Section 4.7.

only reflect a small portion of the rich information in knowledge graphs. Obviously, the hand-designed methods are not suitable for the complex and flexible node importance inferring such as the prediction of an unreleased movie according to its attributes and its links in the knowledge graph.

To alleviate this problem, some machine-learning based methods have been recently proposed such as GENI [17] and MULTIIMPORT [18]. GENI maps the node features obtained by node2vec [9] to importance values, and adaptively aggregates these values from different edge types in a manner of the graph attention network. Benefiting from the supervised learning framework and the graph attention mechanism, GENI achieves higher performance than the hand-designed methods. On the basis of GENI, MULTIIMPORT further explores how to learn the latent node importance with external input importance signals. Despite their success in accurate node importance estimation, they omits the node descriptions that is potentially related to node importance. Moreover, modeling the complicate interaction between nodes directly from scalars may limit the representation ability of graph neural networks. Making full use of the graph topology and the node semantic contents with a delicate model design becomes the key challenge to infer node importance.

In this paper, we propose a representation learning based framework to leverage the rich information comprehensively for this problem. Specifically, we provide the **R**elational **G**raph **T**ransformer **N**etwork (RGTN), where a relational graph transformer is proposed firstly to propagate the node information with the regard of semantic predicate representations, under the assumption that different predicates may have distinct effects on the transmission of node importance. Then, we take two separate transformer encoders to capture the structural and semantic information of nodes respectively, and develop a co-attention module to fuse the two separate representations of the nodes flexibly. Next, an attention-based aggregation module is adopted to map the representations of nodes to their importance values. In addition, we introduce a learning-to-rank loss to learn the representations which are aware of the relative ranking information among nodes. Finally, we have conducted extensive experiments on real-world knowledge graphs, and the results illustrate that our model outperforms the existing methods consistently among all the metrics. Our contributions are summarized as follows:

- This paper provides a representation learning based framework for node importance estimation, which could comprehensively leverage the abundant structural and semantic information contained in the various types of nodes and relations.
- A relational graph transformer is designed to propagate the embedding of nodes with the learnable predicate representations, which could differentiate the impacts of predicate semantics on node importance transmission.
- A learning-to-rank loss is introduced to guide the representation learning process, so as to obtain relative-position-aware node representations, which could improve the overall quality of node importance estimation.

2 PRELIMINARIES

This section provides the definitions of key concepts and the formalization of the research problem.

2.1 Definitions

Definition 2.1. **Knowledge Graph.** A Knowledge Graph is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$ with an edge type mapping function $\phi : \mathcal{E} \rightarrow \mathcal{P}$, where nodes $\mathcal{V}$, edges $\mathcal{E}$, edge types $\mathcal{P}$ correspond to entities, relations and predicates. Each edge belongs to a unique predicate. Unlike the homogeneous graphs where nodes are linked with just one edge type, there could be multiple different predicates between two entities.

Example. As shown in Figure 1, a movie knowledge graph contains multiple entities (*e.g.* , "Guy Ritchie" is a person entity and "The Gentlemen" is a movie entity) and multi-relations (*e.g.* , the entity "Matthew McConaughey" has the relation of "starred-in" and "known-for" with the entity "Dallas Buyers Club").

Definition 2.2. **Node Importance.** A node importance $s \in \mathbb{R}^+$ is a non-negative real number represents the significance or the popularity of an entity in a knowledge graph. For instance, the gross of the movie or the voting number for the movie on the website can be regarded as the node importance in movie knowledge graphs. The specific importance value of a node is collected from the real scenarios and obtained after the log transformation.

Example. In Figure 1, the number of votes for the entity "The Gentlemen" is *194,852*, and it would be the node importance value with the log transformation.

Definition 2.3. **Node Description.** A node description is a natural language text, t, which specifically represents the semantic information of the entity. Node-specific descriptions are often rich in many scenarios, which can make the knowledge graphs further informative.

Example. In Figure 1, the movie plot linked to "The Gentlemen" is the node description for this entity, and the person biography linked to "Guy Ritchie" is another type of node description.

2.2 Problem Formalization

Given a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{P})$, a set of node descriptions $\{t\}$, and a set of node importance values $\{s\}$, node importance estimation aims to learn a function $f : \mathcal{V} \rightarrow \mathbb{R}$ that predicts the importance values for each entity in knowledge graphs.

Node importance estimation could serve many potential applications. As in movie knowledge graphs, we can predict the importance of newly added movie entities to help optimize resource allocation. We can also change the links between a movie and different actors to obtain various importance values as a reference for the movie cast selection.

3 METHODOLOGY

In this section, we first present our proposed framework, and then shows the details of each component.

3.1 Framework

The proposed framework is shown in the left side of Figure 2. As discussed in [26], the GNN-based methods could not always capture the graph topology and node semantic information well at the same time in some cases. In addition, the structure and semantic information may contribute differently for various nodes in node importance estimation. Therefore, we apply two separate graph transformer encoders to encode the structural and semantic representations of nodes respectively to fully enjoy the benefits of both. These two graph encoders are composed of relational graph transformer layers, the details of which are shown in the right side of Figure 2. Each encoder deals with a relational graph copied from the original graphs. These graphs have the same nodes and edges, but the different node attribute features. Following [17], the node structure features are obtained from the node2vec [9], which aims to capture the co-occurrence of the adjacent nodes and provides the greater receptive filed in graph for the GNN encoders. As for the node descriptions, Transformer-XL [6] is used to embed the text into the node semantic features. After passing through two graph encoders, a co-attention fusion mechanism is adopted to interact the structure-specific features and the semantic-specific features. Then these features are respectively projected into importance values, and aggregated with attention weights to get the final importance values of the nodes. At last, the Learning-To-Rank (LTR) loss and the Root Mean Square Error (RMSE) are used to train the whole model.

3.2 Relational Graph Transformer

To infer the node importance in knowledge graphs accurately, it is not only necessary to consider the attribute features of the target nodes themselves, but also valuable to leverage the information of neighbor nodes and consider their predicate types simultaneously. We design a *Relational Graph Transformer* stacking L relational graph transformer layers (shown in the right side of Figure 2) to handle the complex multi-relational graphs, and generate the informative embeddings for the subsequent modules.

We first make two copies for the input knowledge graph as shown in Figure 2, one using the features $X^{(s)}$ from node2vec that focus on graph topology as the node initialization features, and the other using the textual features $X^{(c)}$ encoded by Transformer-XL that focus on node semantics as the initialization. The predicate features are initialized randomly, and denoted as P. Two separate encoders handle these two types of features. The detailed explanation of the relational transformer layer is as follows.

Given a target node v, we aim to aggregate the messages of the first-order neighbor nodes (defined as the source nodes u) in the manner of self-attention [23]. We define the h_v^l and h_u^l as the feature of the target node and the source node at the l-th layer with the dimension d, and we set the node original semantic features $X^{(c)}$ or structure features $X^{(s)}$ as the input features h^0 for the first layer. Then the weights for message aggregations are calculated by

$$w_{(u,v),m}^{k,l} = \frac{W_Q^{k,l}\, h_v^l \cdot W_K^{k,l}\, h_u^{l\,T}}{\sqrt{d}} \cdot W_E^{k,l}\, P_{(u,v)}^m \ , \tag{1}$$

$$a_{(u,v)}^{k,l} = \sum_m \frac{exp(w_{(u,v),m}^{k,l})}{\sum_{u' \epsilon \mathcal{N}_v} \sum_{m'} exp(w_{(u',v),m'}^{k,l})} \ , \tag{2}$$

where $W_Q^{k,l}$, $W_K^{k,l}$ and $W_E^{k,l}$ denote the projection matrices for the k-th head attention, and $\mathcal{N}_v$ is the list of neighbor nodes of the node v. $P_{u,v}^m$ denotes the trainable predicate embedding for the m-th edge between the node u and the node v. We explicitly add the predicate representations to the message passing in Equation (1), based on the assumption that different predicates may have distinct effects on the transmission of importance. The total message for the node v is then aggregated by

$$M_v^l = W_o^l \ \|_{k=1}^H (\sum_{u \epsilon \mathcal{N}_v} a_{(u,v)}^{k,l} \cdot W_V^{k,l} h_v) \ , \tag{3}$$

where W_o and W_V are the transformation matrices, H is the total number of attention heads and $(\|\cdot)$ is the vector concatenation operation. Similar to the classical Transformer architecture, the M_v^l are passed to the Feed Forward Network (FFN) and normalization layers with residual connections, as:

$$h_v^{l\,\prime} = Norm(h_v^l + M_v^l) \ , \tag{4}$$

$$h_v^{l\,\prime\prime} = FFN(h_v^{l\,\prime}) \ , \tag{5}$$

$$h_v^{l+1} = Norm(h_v^{l\,\prime} + h_v^{l\,\prime\prime}) \ , \tag{6}$$

where $FFN(\cdot)$ denotes a network with two linear layers and a *ReLU* activation layer, and $Norm(\cdot)$ is the normalization layer like Batch Normalization (BN) or Layer Normalization (LN). The last layer output of the relational graph transformer encoder on structure information is regarded as the node structure representation $h^{(s)}$, and the output of another encoder on semantic information is regarded as the node content representation $h^{(c)}$.

3.3 Co-attention Fusion Module

The contribution of node structure information and semantic information to the final node importance may be different in distinct scenarios. For example, in a movie knowledge graph, with a prevalent plot design, some movies may become popular even if the main creators are less famous. The opposite situation is that with several maestros participating in, some movies are likely to be paid attention to even if the plots look unpopular. Therefore, fusing the representations of nodes from structure and semantic information adaptively would benefit the performance of node importance

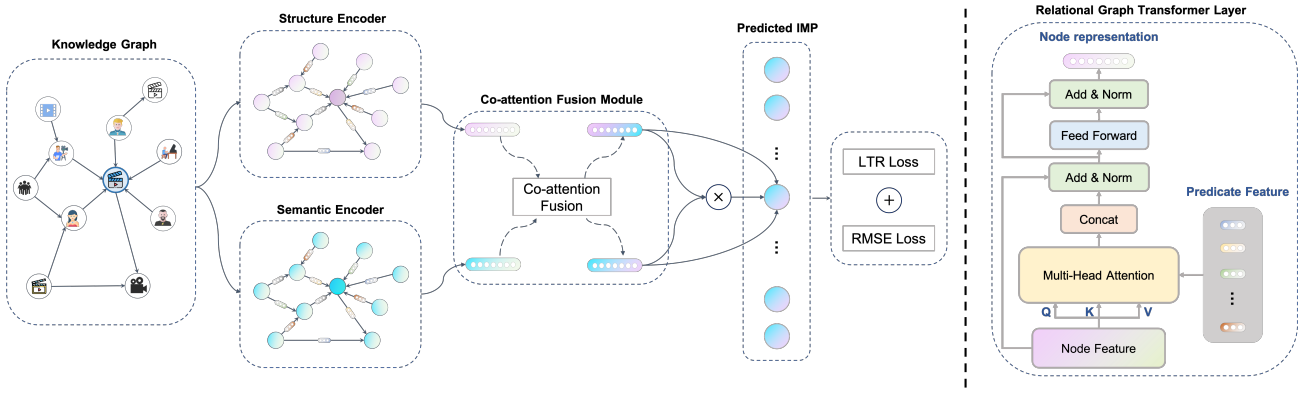

Figure 2: The framework of the proposed model (left). Given a knowledge graph, extract the structure features (in pink) and the semantic features (in blue) as node attribute features and send them to the corresponding relational graph transformer encoders. The co-attention fusion module fuses the two separate node representations, and predicts the importance values (IMP). The LTR loss ensures that the learned representations can be aware of the relative ranking information among nodes. The right is the detail of the relational graph transformer layer, which is the component of the structure encoder and semantic encoder.

estimation. For the target node v, we obtain its structure representation $h_v^{(s)}$ and its semantic representation $h_v^{(c)}$. Since these representations contain different properties of the node, we apply a co-attention fusion module to enhance them by interacting with each other. We first make a transformation for the features via a matrix W_F, as:

$$z_v^1 = W_F h_v^{(s)} \ , \ z_v^2 = W_F h_v^{(c)} \ . \tag{7}$$

The fusion for the two aspect representation is calculated by

$$\beta_{i,j} = \frac{exp(W_Q{}' z_v^i \cdot W_K{}' z_v^{j^T})}{\sum_{j' \epsilon \{1,2\}} exp(W_Q{}' z_v^i \cdot W_K{}' z_v^{j'^T})} \ , \tag{8}$$

$$\hat{h_v^{(s)}} = Norm(FFN'(\beta_{1,1} z_v^1 + \beta_{1,2} z_v^2) + h_v^{(s)}) \ , \tag{9}$$

$$\hat{h_v^{(c)}} = Norm(FFN'(\beta_{2,1} z_v^1 + \beta_{2,2} z_v^2) + h_v^{(c)}) \ , \tag{10}$$

where $W_Q{}'$ and $W_K{}'$ are the matrices with the same shape of W_F. $\hat{h_v^{(s)}}$ and $\hat{h_v^{(c)}}$ are the enhanced representations corresponding to the node v.

3.4 Importance Prediction Module

After the co-attention fusion module, we project two informative embeddings into 1-dimensional importance values and combine them with attention weights. The weights are calculated by

$$\gamma^{(s)} = \frac{exp(\hat{h_v^{(s)}} \lambda^T)}{exp(\hat{h_v^{(s)}} \lambda^T) + exp(\hat{h_v^{(c)}} \lambda^T)} \ , \quad \gamma^{(c)} = \frac{exp(\hat{h_v^{(c)}} \lambda^T)}{exp(\hat{h_v^{(s)}} \lambda^T) + exp(\hat{h_v^{(c)}} \lambda^T)} \ , \tag{11}$$

where λ denotes the trainable attention vector. The final importance is obtained by

$$s_v^{(s)} = W_s^1 \hat{h_v^{(s)}} \ , \ s_v^{(c)} = W_s^2 \hat{h_v^{(c)}} \ , \tag{12}$$

$$s_v^* = LeakyReLU(\gamma^{(s)} s_v^{(s)} + \gamma^{(c)} s_v^{(c)}) \ , \tag{13}$$

where W_s^1 and W_s^2 are the matrices for the projection. The ultimate importance value for the node v is s_v^*, which captures both structure and semantic information thoroughly.

3.5 Learning-to-rank Loss

The Learning-To-Rank (LTR) loss takes importance values of all nodes as input. From the perspective of an entire knowledge graph, we are concerned about the importance rankings among nodes. However, the ranking information of nodes is ignored in the learning process of previous NIE models. We introduce the listwise LTR loss [5] for this task. The additional benefit brought by the LTR loss is to jump out of the locality of GNN's node embedding, and therefore to learn the distributions of node importance from a global view. At the same time, the softmax function introduces the competitiveness among nodes, which could promote the model to learn better representations. We sample n nodes for the node v to form a node set $\mathcal{N}_v^{(r)}$. The LTR loss are calculated as follows:

$$s_v{}' = \frac{exp(s_v)}{\sum_{j \epsilon \mathcal{N}_v^{(r)}} exp(s_j)} \ , \ s_v^{*'} = \frac{exp(s_v^*)}{\sum_{j \epsilon \mathcal{N}_v^{(r)}} exp(s_j^*)} \ , \tag{14}$$

$$\mathcal{L}_v^{(r)} = - \sum_{i \epsilon \mathcal{N}_v^{(r)}} s_i{}' log(s_i^{*'}) \ , \tag{15}$$

where s_v is the ground truth importance value for the node v and s_v^* is the predicted one.

3.6 Training Process

Root Mean Square Error (RMSE) is the commonly used regression loss. It is defined as

$$\mathcal{L}_0 = \frac{1}{|\mathcal{V}_s|} \sum_{i \epsilon \mathcal{V}_s} (s_i^* - s_i)^2 \ , \tag{16}$$

where $\mathcal{V}_s$ is the set of nodes with the valid ground truth importance values. For the stability of training, we also add two auxiliary RMSE

loss for the $s^{(s)}$ and $s^{(c)}$ as follows,

$$\mathcal{L}_1 = \frac{1}{|\mathcal{V}_s|} \sum_{i \epsilon \mathcal{V}_s} (s_i^{(s)} - s_i)^2 \ , \ \mathcal{L}_2 = \frac{1}{|\mathcal{V}_s|} \sum_{i \epsilon \mathcal{V}_s} (s_i^{(c)} - s_i)^2 \ . \quad (17)$$

We optimize the proposed model by minimizing the RMSE loss and the LTR loss together. One allows the model to learn the importance value, and the other focuses on the ranking relationship of node importance. The total loss is calculated by

$$\mathcal{L} = a\mathcal{L}_0 + b(\mathcal{L}_1 + \mathcal{L}_2)/2 + c(\frac{1}{|\mathcal{V}_s|} \sum_{j \epsilon \mathcal{V}_s} \mathcal{L}_j^{(r)}) \ , \quad (18)$$

where a, b, c are the hyper-parameters to control the ratio of different parts of the loss.

4 EXPERIMENTS

In this section, we construct extensive experiments on real-world data, and aim to explore the following questions.

- How does RGTN perform on real-world knowledge graphs, compared with the existing baselines and the previous state-of-the-art model?
- How do the components of RGTN affect node importance estimation? In particular, how well do the proposed framework improve with other graph encoders?
- How effective is RGTN's representation learning?

4.1 Dataset Description

We conduct experiments on three public knowledge graphs with different properties, which are used in previous node importance study [17, 18]. We build up these datasets following the description in [17], but there are some slight differences as [17] does not specify the predicate lists of the knowledge graphs. A brief description of datasets is given below. More details are shown in Table 1.

- **FB15K** [2] is a subset of FreeBase [1], which contains rich general knowledge base relation triples and texual information of entity pairs. The characteristic of FB15K is its significantly larger number of predicates than other graphs we evaluated. For each entity in the graph, we consider the pageview number in last 30 days of the corresponding Wikipedia page as the node importance, and take the wikidata's description of the entity as the node semantic information.
- **IMDB** is a moive knowledge graph generated from the IMDB dataset[1], which consists of the entities of movies, genres, casts, crews, publication company and countries. The vote number for movies provided by the IMDB dataset is used as the node importance. As for the node descriptions, we collect the plot summaries of movies and personal biographies of people to enrich the graph information. It is the largest knowledge graph that we used.
- **TMDB5K** is another movie knowledge graph including movie entities and other related entities like actors, casts, crews and companies. TMDB is built from TMDB 5000 dataset[2]. The node importance is defined by the movie popularity score, and the semantic information of nodes comes from the movie overviews.

[1]https://www.imdb.com/interfaces/
[2]https://www.kaggle.com/tmdb/tmdb-movie-metadata

Table 1: Real-world datasets summary. PRED: predicate number. Node w/ IMP: number of nodes with importance.

Name	Nodes	Edges	PRED	Nodes w/ IMP
FB15K	14,951	592,213	1,345	14,105(94%)
IMDB	1,124,995	9,729,868	30	202,538(18%)
TMDB5K	114,805	761,648	34	4,803(4%)

4.2 Baseline Methods

We consider the following methods for comparison, which could be roughly divided into three families of algorithms.

Random walk based approaches. PageRank (PR) [16] and Personalised PageRank (PPR) [10] are representative random walk based algorithms to utilize the graph topology for node importance inferring.

Non-graph supervised approaches. We choose several classical supervised methods to examine their performance on node importance inferring, including Linear Regression (LR), Random Forests (RF), MultiLayer Perceptron (MLP). These methods ignore the graph structure and focus on the node attribute features primarily.

Graph neural network based approaches. Graph neural networks have shown great potentials in processing the graph data in term of representation learning. We explore the effectiveness of these methods on this problem:

- Graph Convolution Network (GCN) [13]: GCN performs graph convolutions in the Fourier domain by aggregating the neighbor node features.
- Graph ATtention network (GAT) [24]: GAT designs the multi-head attention mechanism for the GNN's aggregation by assigning different weights to the neighbors adaptively.
- Relational Graph Convolution Network (RGCN) [20]: RGCN introduces specialized transformation matrices for each type of relations, in order to handle complex edge relations in knowledge graphs.
- Composition-based multi-relational Graph Convolution Network (CompGCN) [22]: CompGCN leverages the entity-relation composition operation from knowledge graph embedding techniques to jointly embed both nodes and relations of graphs.
- GNN for Estimating Node Importance (GENI) [17]: GENI applies an attentive GNN for predicate-aware score aggregation to capture relations between neighbor nodes.

4.3 Evaluation Metrics

Following [17], to make a comprehensive evaluation on ranking quality and importance correlation, we use the metrics: Normalized Discounted Cumulative Gain (NDCG) and Spearman's rank correlation coefficient (SPEARMAN). In addition, we define another metric, Overlap (OVER), to reflect the recall of important nodes after ranking node importance. Only the nodes with the valid ground truth importance are taken into evaluation particularly. The formal definitions of metrics are given below.

NDCG is a widely used measure of ranking quality considering the order of elements. In this task, we define the graded relevance

Table 2: Experimental results of different methods over the three real-world datasets. The methods with an asterisk(∗) use only the structure features. The highest results are in bold, and the second highest results are underlined.

Method	FB15K			IMDB			TMDB5K		
	NDCG@100	SPEARMAN	OVER@100	NDCG@100	SPEARMAN	OVER@100	NDCG@100	SPEARMAN	OVER@100
PR	0.8408±0.009	0.3500±0.019	0.1380±0.016	0.8813±0.021	0.1933±0.004	0.3820±0.019	0.8388±0.026	0.6284±0.013	0.4120±0.060
PPR	0.8423±0.009	0.3641±0.021	0.1360±0.014	0.9240±0.008	0.5019±0.005	0.4240±0.016	0.8535±0.008	0.7256±0.011	0.4200±0.052
LR	0.8974±0.010	0.6774±0.013	0.2280±0.033	0.9484±0.002	0.6107±0.004	0.4660±0.031	0.8502±0.014	0.6907±0.012	0.4260±0.030
RF	0.9160±0.009	0.6432±0.008	0.2240±0.043	0.9515±0.004	0.6102±0.003	0.4640±0.012	0.8703±0.011	0.7186±0.005	0.4660±0.046
MLP	0.9201±0.014	0.6916±0.007	0.2940±0.074	0.9437±0.004	0.6306±0.005	0.4320±0.020	0.8767±0.009	0.6841±0.013	0.4840±0.024
GCN	0.9419±0.007	0.7774±0.005	0.4260±0.042	0.9223±0.010	0.7211±0.006	0.3400±0.039	0.9057±0.007	0.7791±0.010	0.5240±0.025
GAT	0.9369±0.011	0.7631±0.016	0.3980±0.055	0.9486±0.007	0.7024±0.059	0.4820±0.029	0.9071±0.010	0.7724±0.007	0.5380±0.029
RGCN	0.8800±0.010	0.7169±0.012	0.2900±0.041	0.9627±0.003	0.7736±0.002	0.5760±0.039	0.9011±0.010	0.7955±0.010	0.5680±0.024
CompGCN	0.9309±0.009	0.7732±0.003	0.4000±0.033	0.9646±0.004	0.7708±0.005	0.5720±0.036	0.8995±0.019	0.7890±0.011	0.5440±0.031
GENI*	0.9308±0.010	0.7799±0.013	0.4320±0.070	0.9496±0.005	0.7371±0.007	0.5180±0.017	0.9124±0.003	0.7943±0.019	0.5580±0.028
GENI	0.9422±0.005	0.7808±0.020	0.4240±0.048	0.9575±0.005	0.7686±0.005	0.5500±0.022	0.9074±0.006	0.7822±0.009	0.5320±0.027
RGTN	**0.9555±0.010**	**0.8204±0.009**	**0.4920±0.068**	**0.9753±0.003**	**0.7906±0.002**	**0.6460±0.036**	**0.9156±0.010**	**0.7965±0.009**	**0.5700±0.038**

values as the ground truth importance values after log transformation. Given a list of nodes with predicted importance S^* and their ground truth importance S, we sort the nodes by the predicted importance and take the corresponding ground truth importance at the position i as rel_i . $DCG@k$ is defined as

$$DCG@k = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1))} \quad . \tag{19}$$

The Ideal DCG (IDCG) is attained by sorting nodes according to their ground truth importance and calculated DCG for this ranked list. Normalized DCG at position k ($NDCG@k$) is calculated by

$$NDCG@k = \frac{DCG@k}{IDCG@k} \quad . \tag{20}$$

SPEARMAN measures the correlation between the node importance list S^* and the ground truth list S. After converting the raw values S^* and S into the ranks R_{s^*} and R_s, $SPEARMAN$ is calculated by

$$SPEARMAN = \frac{cov(R_{s^*}, R_s)}{\sigma_{R_{s^*}} \sigma_{R_s}} \quad , \tag{21}$$

where $cov(R_{s^*}, R_s)$ is the covariance of the rank variables, $\sigma_{R_{s^*}}$ and σ_{R_s} are the standard deviations of the rank variables.

OVER is the overlap ratio of the predicted important nodes and the real important nodes. In detail, we define the set of the top k important predicted nodes as S^*_{top-k} , and the set of nodes with the top k real importance as S_{top-k} . The $OVER@k$ is attained by

$$OVER@k = \frac{\left|S^*_{top-k} \bigcap S_{top-k}\right|}{k} \quad . \tag{22}$$

4.4 Inferring Node Importance on KGs

We benchmark the performance of RGTN on node importance estimation against several competitive models. We split the datasets into training, validation and testing part with the ratio of 7 : 1 : 2, and conduct 5-fold cross validation on three metrics, and report the average and standard deviation values. We set the dimension of node representation to 64, and use Adam [12] optimizer to optimize all the models.

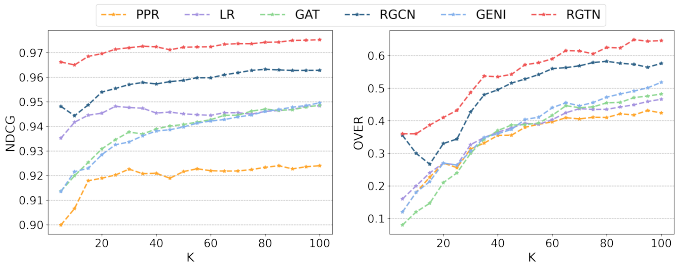

Figure 3: RTGN consistently achieves the best performance when we change the K value from 5 to 100.

For fair comparison, the semantic features and the structure features are concatenated as the node input features, expect for GENI* using the structure features only following [17]. The evaluation results are reported in Table 2.

We summarize some observations after analyzing the results: (1) The proposed RGTN outperforms all the compared models in all metrics. It is noteworthy that RGTN obtains much higher performance on FB15K and IMDB, which are more complicate than TMDB5K. Especially, for OVER@100, RGTN achieves large relative improvements of 13.89% on FB15K and 12.15% on IMDB. The results on real-world datasets demonstrate the effectiveness of RGTN. (2) The relational graph neural network like RGCN and CompGCN could achieve performance comparable to or exceeding GENI, which makes specific designs for node importance inferring such as centrality scaling. GENI prematurely simplified the node features to scalars, making it hard to model the interaction between nodes effectively. Through the representation of the edge relations and the high-order node information, the power of the represetation learning is shown. (3) The experiment results are in line with our intuition that the supervised trainable methods can fit the distribution of node importance more flexibly and accurately. A simple linear regression could also achieve better performance than the

Table 3: The results of RGTN and its variants on FB15K and IMDB dataset. NDCG: NDCG@100. SPM: SPEARMAN. OVER: OVER@100

Data	Metric	RGTN-sem	RGTN-str	RGTN-cat	RGTN-att	RGTN
FB15K	NDCG	0.950±0.01	0.953±0.01	0.951±0.01	0.954±0.01	**0.956±0.01**
	SPM	0.792±0.02	0.808±0.01	0.807±0.01	0.819±0.01	**0.820±0.01**
	OVER	0.450±0.07	0.484±0.04	0.436±0.07	0.436±0.06	**0.492±0.07**
IMDB	NDCG	0.966±0.00	0.961±0.00	0.967±0.01	0.970±0.00	**0.975±0.00**
	SPM	0.782±0.00	0.752±0.00	0.789±0.00	0.790±0.00	**0.791±0.00**
	OVER	0.594±0.05	0.536±0.03	0.592±0.04	0.608±0.05	**0.646±0.04**

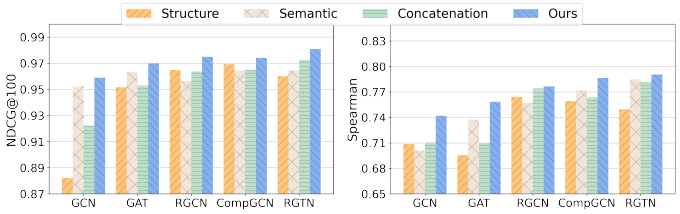

Figure 4: The results of different encoders equipped with the proposed framework. Structure: the models using only structure features. Semantic: the models using only semantic features. Concatenation: the models using both structure and semantic features.

PageRank algorithm on the whole. In addition, compared to MLP, the graph neural network methods take node neighbor information into consideration explicitly, and attain marginal improvement, especially in the *SPEARMAN* metric.

In order to explore the performance of our method comprehensively, we show the RGTN performance changes brought by varoius top-K values, as well as the comparison with other models. As displayed in Figure 3, RGTN outperforms other competitive methods consistently when the K changes from 5 to 100. Therefore, our model could provide accurate node importance estimate for top nodes with different importance levels.

4.5 Component Analysis

We compare RGTN with its four variants on FB15K and IMDB datasets to validate the effectiveness of components in RGTN. The variants are denoted as follows:

- RGTN-sem: RGTN uses the semantic features only with the RMSE loss.
- RGTN-str: RGTN uses the structure features only with the RMSE loss.
- RGTN-cat: RGTN uses the concatenation of semantic and structure features with the RMSE loss.
- RGTN-att: RGTN uses the co-attention fusion mechanism with the RMSE loss.

From the results in Table 3, we can make the following conclusions: (1) The results of RGTN consistently outperform than other variants in two datasets, verifying the advantage of using all components. (2) RGTN-att is generally better than RGCN-cat in all metrics, which indicates the co-attention mechanism fully and

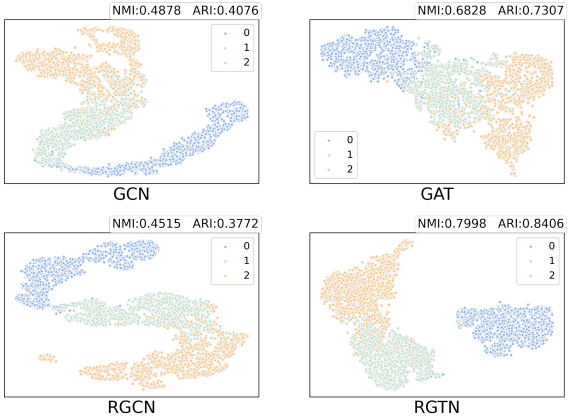

Figure 5: The T-SNE visualization of the subset node embeddings on IMDB dataset.

flexibly utilize the graph topology and node description information. (3) The introduction of the LTR loss improves the NDCG and OVER metrics, demonstrating the performance gain brought by the leaning of relative relationships between the node importance.

To verify the universal effectiveness of the fusion mechanism and the LTR loss in NIE, we combine different graph encoders with these two components, and conduct experiments on the largest dataset IMDB. It could be observed from Figure 4 that the concatenation of features can not utilize the two types of features well, as the semantic or structure encoder may surpass the performance of the concatenated feature encoders in some cases. But our components can boost the evaluation values of these graph encoders on NDCG and SPEARMAN, further reflecting the ability of our methods to improve the node importance estimation.

4.6 Representation Learning

We conduct the node clustering and visualization task to show off the effectiveness of the learned node representations. In detail, we choose a test set of the IMDB dataset with more than 40, 000 nodes. After sorting the nodes in order of importance, we select three types of nodes as three categories: the top 1, 000 nodes, the middle continuous 1, 000 nodes, and the bottom 1, 000 nodes. We obtain the node embeddings on the last layer of models before being projected into scalars, then cluster these representations for ten times using an unsupervised clustering algorithms, k-means. We measure the mean of the Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI) to evaluate clustering quality, and the specific values are tagged in the sub-graphs of Figure 5. Since GENI does not learn the node representation vectors specifically, we exclude it for the comparison in this subsection. RGTN attains the highest values in NMI and ARI against other graph models such as GCN, GAT, and RGCN. For a more intuitive comparison, we plot the learned representations of three categories using t-SNE [21], coloring by the category labels in Figure 5. It can be found that the visualization of RGTN performs best with a more compact cluster structure and the clearest distinct boundaries across categories. We apply another visualization to explore the correlation between the distribution of

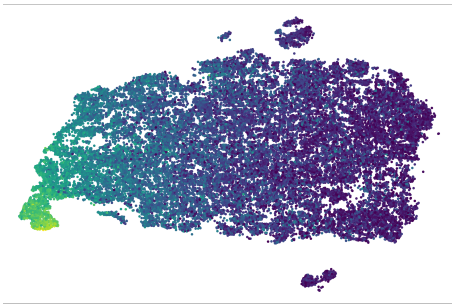

Figure 6: The plots for node representations in IMDB. The brighter color means the higher node importance.

node representations and the variation of node importance from the overall perspective. We project the node representations in IMDB into 2-dimensional space and dye the more important nodes with the brighter colors. In Figure 6, we observe that the nodes with high importance are gathered on the one side of the space, the nodes with low importance are clustered on the other side of the space, and the middle area between the two ends presents a distribution pattern with gradual color change. As t-SNE could ensure that the neighbor points on the Figure 6 are similar in high dimensional space, we believe that RGTN has learned the representation that correlated with the overall node importance closely.

4.7 Application Scenarios

The forecasting task is one of the promising scenarios for node importance estimation, which is concerned with the importance or popularity of the newly added entities in knowledge graphs. Take the movie datasets like IMDB as an example. The movies released earlier than 2015 are used as the training samples, and the movies from 2015 to 2020 are used for importance forecasting. We show the 10 most popular movies since 2015 that our model predicts in Table 4. "Avengers: Endgame" is the most important movie since 2015 according to the predictions, and in reality, this movie has achieved great success. With a global box office of 2.7 billion US dollars, it ranks among the top in the global movie history[3]. The other predicted movies also have relatively high popularity rankings among overall movies. This shows the potential of our model in forecasting scenarios.

Node importance estimation could also be applied to such application scenarios of the film cast selection. Given the plot summary of a movie as the node semantic information, and the basic connections in knowledge graphs, different importance values can be obtained by changing the links between the movie entity and various actor entities, thereby providing a reference for the film director. Node importance estimation is supposed to serve many such potentially interesting data mining applications.

5 RELATED WORK

This section summarizes the existing literature related to this work, which includes node importance estimation methods and graph learning methods.

[3]https://www.boxofficemojo.com/chart/top_lifetime_gross/?ref_=bo_cso_ac

Table 4: The top 10 important movies since 2015 predicted by the proposed model. Vote: the recent votes number in IMDB website. POP: the real-time popularity ranking among all the movies provided by IMDB website.

Ranking	Movie	Year	Vote	POP
1	Avengers: Endgame	2019	810,052	26
2	Guardians of the Galaxy Vol. 2	2017	570,009	166
3	X-Men: Apocalypse	2016	393,044	493
4	Thor: Ragnarok	2017	587,812	126
5	Captain America: Civil War	2016	393,044	493
6	Avengers: Infinity War	2018	834,546	84
7	Star Wars: Episode VIII - The Last Jedi	2017	565,136	256
8	Spider-Man: Homecoming	2017	513,069	202
9	Baby Driver	2017	439,453	193
10	Rogue One	2016	556,633	132

Node Importance Estimation is a crucial research topic in network science, enabling a series of practical applications. The classical algorithm [8] explores the node importance in social network by defining centrality indices to measure the contribution of nodes to the network flow. [3] and [4] further expand and improve the measure of node centrality. Another classical algorithm, PageRank [16], propagates the importance of nodes based on the random walk. Personalized PageRank [10] adjusts the node weights or edge weights to bias the random walk considering specific topics. However, these methods could not handle the variant definitions of node importance on the one hand, and could not capture the rich information such as multiple relations and various node attributes in knowledge graphs on the other hand. Recently, with the development of deep learning on the graph data, GENI [17] is developed to utilize the multi-relational graph structure information to infer the node importance under the formulation of supervised learning. MULTIIMPORT [18] utilizes multiple input signals to supervise the learning of latent node importance. Despite the success of supervised graph neural networks in node importance, the abundant text descriptions for nodes are omitted from the model design. We propose a new representation learning based framework to leverage both the graph topology and the node semantic information.

Graph Neural Networks (GNNs) have attracted tremendous research attention in recent years [13, 15, 24, 28], and have been applied in a series of applications [25, 29, 30]. Kipf and Welling [13] propose a graph convolution model to perform convolutions in the Fourier domain by aggregating the neighbor node features. Velickovic *et al.* [24] adopt the attention mechanism in GNNs to aggregate neighbourhood information adaptively. Xu *et al.* [28] analyze the expressive power of GNNs and develop a simple but expressive architecture in graph classification task. Li *et al.* [15] propose several effective components to facilitate the training of deeper GNNs. These advanced works boost the performance of GNN, but are limited to the study of homogeneous graphs. RGCN [20] introduces specific transformation matrices for different types of edges, in order to handle multi-relational graphs. CompGCN [22] further integrates the entity-relation composition operations with GNNs, jointly embedding both nodes and relations in graphs. As Transformer [23] is the best performing architecture in the filed of natural language processing, several pioneer works [7, 11, 19, 31] have tried to introduce such components in graph domain to

enjoy the powerful ability of the self-attention mechanism. But there is still no specific GNNs with the transformer structure for knowledge graphs. We design the RTGN which collaborates the representation of entities and predicates into the message passing process, effectively using the rich information in knowledge graphs.

6 CONCLUSION

Node importance estimation is a significant task in graph data mining, which could also facilitate many potential application scenarios. A major challenge to infer node importance lies in leveraging the rich structure and semantic information contained in the multifarious nodes and edges. In this paper, we develop a representation learning based framework that adaptively capture the comprehensive characteristics in complex graphs for accurate node importance estimation. The relative ranking information among nodes are also injected into the learning process. We have conducted extensive experiments on real-world knowledge graphs to demonstrate the superiority of our frameworks over previous methods. For the future work, we intend to explore the evolution patterns of node importance, that is, how the importance of nodes changes in dynamic knowledge graphs.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under grants 71901011, 51822802, 51778033, 51991395 and U1811463.

REFERENCES

[1] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*. ACM, 1247–1250.

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. 2787–2795.

[3] Stephen P. Borgatti. 2005. Centrality and network flow. *Soc. Networks* 27, 1 (2005), 55–71.

[4] Stephen P. Borgatti and Martin G. Everett. 2006. A Graph-theoretic perspective on centrality. *Soc. Networks* 28, 4 (2006), 466–484.

[5] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007 (ACM International Conference Proceeding Series, Vol. 227)*. ACM, 129–136.

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2978–2988.

[7] Vijay Prakash Dwivedi and Xavier Bresson. 2020. A Generalization of Transformer Networks to Graphs. *CoRR* abs/2012.09699 (2020).

[8] Linton C Freeman. 1978. Centrality in social networks conceptual clarification. *Social networks* 1, 3 (1978), 215–239.

[9] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 855–864.

[10] Taher H. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA*. ACM, 517–526.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 2704–2710.

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[14] Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J.ACM* 46, 5 (1999), 604–632.

[15] Guohao Li, Chenxin Xiong, Ali K. Thabet, and Bernard Ghanem. 2020. DeeperGCN: All You Need to Train Deeper GCNs. *CoRR* abs/2006.07739 (2020).

[16] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[17] Namyong Park, Andrey Kan, Xin Luna Dong, Tong Zhao, and Christos Faloutsos. 2019. Estimating Node Importance in Knowledge Graphs Using Graph Neural Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 596–606.

[18] Namyong Park, Andrey Kan, Xin Luna Dong, Tong Zhao, and Christos Faloutsos. 2020. MultiImport: Inferring Node Importance in a Knowledge Graph from Multiple Input Signals. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 503–512.

[19] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. 2020. Self-Supervised Graph Transformer on Large-Scale Molecular Data. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[20] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10843)*. Springer, 593–607.

[21] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[22] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008.

[24] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[25] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM, 165–174.

[26] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. 2020. AM-GCN: Adaptive Multi-channel Graph Convolutional Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1243–1253.

[27] Biao Xiang, Qi Liu, Enhong Chen, Hui Xiong, Yi Zheng, and Yu Yang. 2013. PageRank with Priors: An Influence Propagation Perspective. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2740–2746.

[28] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[29] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. ACM, 974–983.

[30] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. ijcai.org, 3634–3640.

[31] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2019. Graph Transformer Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 11960–11970.

A APPENDIX

In the appendix, some details of the datasets and experiments are introduced.

A.1 Extra Details of Datasets

FB15K. We use the triplets data provided by the original FB15K dataset, and collect the node descriptions from WikiData[4]. We first map the entity ids to the corresponding entity name, and take the description of entities in WikiData as the node semantic information. Some examples are shown Table 5.

Table 5: Description Examples in FB15K

MID	Descriptions
/m/01g2q	Bipolar disorder, human mental illness characterized by mood changes
/m/0nv6n	McHenry County Illinois, county in Illinois, United States
/m/027jq2	Emir Kusturica, Serbian film director, actor and musician of Bosnian origin
/m/072zl1	Pride & Prejudice (2005 film) , 2005 period film directed by Joe Wright

IMDB. We use the entities in IMDB Dataset including movies, persons (principal casts and crews) and movie genres. We define 15 predicates and their corresponding reversed predicates, totaling 30 predicates. The specific predicates we chosen are: "is a", "known for", "belong to the genre of", "as actress", "as actor", "as director", "as writer", "as cinematographer", "as producer", "as composer", "as production designer", "self", "as editor", "do archive footage" and "do archive sound". We collect the plot summaries via IMDB api as the semantic information of movie entities, and person biographies for person entities, shown in Table 6. Those nodes that have no corresponding text descriptions are usually uncommon entities and are not included in this knowledge graph.

TMDB5K The entities in TMDB5k including movies, movie genres, companies, countries, crews and casts. We define 17 predicates and their reversed predicates: "is a", "belong to the genre of" , "published by the company of", "released in the country of", "as actor", "as art crew", "as sound crew" , " as costume and make-up crew", "as crew for actors", "direct", "edit", "as normal crew", "as lighting crew", "as visual effect crew", "as production crew", "as camera crew" and "write". The movie overviews provided by original dataset are considered to the semantic information. Table 7 shows the examples of movie overviews.

A.2 Experiment Details

Feature Extraction. We extract the structure features using the node2vec implemented by pytorch geometric[5]. The dimension of the structure feature is set to 64 in FB15K and TMDB5K, and set to 128 in IMDB. The semantic features are extracted by the pretrained model of Transformer-XL[6] with 768 dimensions. The nodes without semantic information take the mean of neighbor node features as their semantic features.

Training Setting. We train the model in IMDB with the manner of mini-batch. The batch size is set to 2048, and the learning rate is set to 0.001. The models are trained 200 epochs in total with the early stopping of 50 epochs. As for FB15K and TMDB5K, we train the model with the manner of full-batch, while the learning rate is set to 0.005 and the total epochs are up to 5000 with early stopping.

Hyper-parameters Setting. We stack 2 layer for RGTN with 8 attention heads, and the dimension of hidden features is set to 64. The hyper-parameters that control the loss composition, a , b and c are set to 0.3, 0.7, 0.5 for all datasets. And the number of nodes to calculate the LTR loss are chosen from 20, 50, 100, 200.

Baseline Implement. For PR and PPR, we use the default values from NetworkX's *pagerank_scipy* function. For LR, RF and MLP, we use default parameters for the model provided by *scikit-learn*. And the graph models are modified from the DGL examples.

Table 6: Description Examples in IMDB

ID	Descriptions
tt8357692	In rural villages of India, a child born with a specific birth defect may be transformed overnight into a Hindu God becoming a beacon of hope. One boy never relished in being seen as a Hindu God, while the other only knows how to live as a symbol of hope to his village. Both were never given the choice.
tt8367814	An American expat tries to sell off his highly profitable marijuana empire in London, triggering plots, schemes, bribery and blackmail in an attempt to steal his domain out from under him.
nm0005363	Guy Ritchie was born in Hatfield, Hertfordshire, UK on September 10, 1968. After watching Butch Cassidy and the Sundance Kid (1969) as a child, Guy realized that what he wanted to do was make films. He never attended film school, saying that the work of film school graduates was boring and unwatchable. ……
nm0000190	American actor and producer Matthew David McConaughey was born in Uvalde, Texas. His mother, Mary Kathleen (McCabe), is a substitute school teacher originally from New Jersey. His father, James Donald McConaughey, was a Mississippi-born gas station owner who ran an oil pipe supply business. ……

Table 7: Description Examples in TMDB5k

ID	Descriptions
0	In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization.
1	Captain Barbossa, long believed to be dead, has come back to life and is headed to the edge of the Earth with Will Turner and Elizabeth Swann. But nothing is quite as it seems.
2	A cryptic message from Bond's past sends him on a trail to uncover a sinister organization. While M battles political forces to keep the secret service alive, Bond peels back the layers of deceit to reveal the terrible truth behind SPECTRE.

[4]http://wikidata.org/
[5]https://github.com/rusty1s/pytorch_geometric/blob/master/torch_geometric
[6]https://github.com/huggingface/transformers