Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs

Deepak Nathani* **Jatin Chauhan*** **Charu Sharma*** **Manohar Kaul**
Department of Computer Science and Engineering, IIT Hyderabad
{deepakn1019,chauhanjatin100,charusharma1991}@gmail.com,
mkaul@iith.ac.in

Abstract

The recent proliferation of knowledge graphs (KGs) coupled with incomplete or partial information, in the form of missing relations (links) between entities, has fueled a lot of research on knowledge base completion (also known as relation prediction). Several recent works suggest that convolutional neural network (CNN) based models generate richer and more expressive feature embeddings and hence also perform well on relation prediction. However, we observe that these KG embeddings treat triples independently and thus fail to cover the complex and hidden information that is inherently implicit in the local neighborhood surrounding a triple. To this effect, our paper proposes a novel attention-based feature embedding that captures both entity and relation features in any given entity's neighborhood. Additionally, we also encapsulate relation clusters and multi-hop relations in our model. Our empirical study offers insights into the efficacy of our attention-based model and we show marked performance gains in comparison to state-of-the-art methods on all datasets.

1 Introduction

Knowledge graphs (KGs) represent knowledge bases (KBs) as a directed graph whose *nodes* and *edges* represent *entities* and *relations between entities*, respectively. For example, in Figure 1, a triple *(London, capital_of, United Kingdom)* is represented as two entities: *London* and *United Kingdom* along with a relation *(capital_of)* linking them. KGs find uses in a wide variety of applications such as semantic search (Berant et al., 2013; Berant and Liang, 2014), dialogue generation (He et al., 2017; Keizer et al., 2017), and question answering (Zhang et al., 2016; Diefenbach et al., 2018), to name a few. However, KGs typically suffer from missing relations (Socher et al., 2013a; West et al., 2014). This problem gives rise to the task of *knowledge base completion* (also referred to as *relation prediction*), which entails predicting whether a given triple is valid or not.

State-of-the-art relation prediction methods are known to be primarily *knowledge embedding* based models. They are broadly classified as *translational* models (Bordes et al., 2013; Yang et al., 2015; Trouillon et al., 2016) and *convolutional neural network* (CNN) (Nguyen et al., 2018; Dettmers et al., 2018) based models. While translational models learn embeddings using simple operations and limited parameters, they produce low quality embeddings. In contrast, CNN based models learn more expressive embeddings due to their parameter efficiency and consideration of complex relations. However, both translational and CNN based models process each triple independently and hence fail to encapsulate the semantically rich and latent relations that are inherently present in the vicinity of a given entity in a KG.

Motivated by the aforementioned observations, we propose a *generalized* attention-based graph embedding for relation prediction. For node classification, *graph attention networks* (GATs) (Veličković et al., 2018) have been shown to focus on the most relevant portions of the graph, namely the node features in a 1-hop neighborhood. Given a KG and the task of relation prediction, our model *generalizes* and *extends* the attention mechanism by guiding attention to both entity (node) and relation (edge) features in a multi-hop neighborhood of a given entity / node.

Our idea is: 1) to capture multi-hop relations (Lin et al., 2015) surrounding a given node, 2) to encapsulate the diversity of roles played by an entity in various relations, and 3) to consolidate the existing knowledge present in semantically similar relation clusters (Valverde-Rebaza

* Equal Contribution

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4710–4723
Florence, Italy, July 28 - August 2, 2019.

and de Andrade Lopes, 2012). Our model achieves these objectives by assigning different weight mass (attention) to nodes in a neighborhood and by propagating attention via layers in an iterative fashion. However, as the model depth increases, the contribution of distant entities decreases exponentially. To resolve this issue, we use relation composition as proposed by (Lin et al., 2015) to introduce an auxiliary edge between n-hop neighbors, which then readily allows the flow of knowledge between entities. Our architecture is an encoder-decoder model where our *generalized graph attention model* and *ConvKB* (Nguyen et al., 2018) play the roles of an *encoder* and *decoder*, respectively. Moreover, this method can be extended for learning effective embeddings for Textual Entailment Graphs (Kotlerman et al., 2015), where global learning has proven effective in the past as shown by (Berant et al., 2015) and (Berant et al., 2010).

Our contributions are as follows. To the best of our knowledge, we are the first to learn new graph attention based embeddings that specifically target relation prediction on KGs. Secondly, we generalize and extend graph attention mechanisms to capture both entity and relation features in a multi-hop neighborhood of a given entity. Finally, we evaluate our model on challenging relation prediction tasks for a wide variety of real-world datasets. Our experimental results indicate a clear and substantial improvement over state-of-the-art relation prediction methods. For instance, our attention-based embedding achieves an improvement of 104% over the state-of-the-art method for the Hits@1 metric on the popular *Freebase (FB15K-237)* dataset.

The rest of the paper is structured as follows. We first provide a review of related work in Section 2 and then our detailed approach in Section 3. Experimental results and dataset descriptions are reported in Section 4 followed by our conclusion and future research directions in Section 5.

2 Related Work

Recently, several variants of KG embeddings have been proposed for relation prediction. These methods can be broadly classified as: (i) compositional, (ii) translational, (iii) CNN based, and (iv) graph based models.

RESCAL (Nickel et al., 2011), NTN (Socher et al., 2013b), and the Holographic embedding model (HOLE) (Nickel et al., 2016) are examples of compositional based models. Both RESCAL and NTN use tensor products which capture rich interactions, but require a large number of parameters to model relations and are thus cumbersome to compute. To combat these drawbacks, HOLE creates more efficient and scalable compositional representations using the circular correlation of entity embeddings.

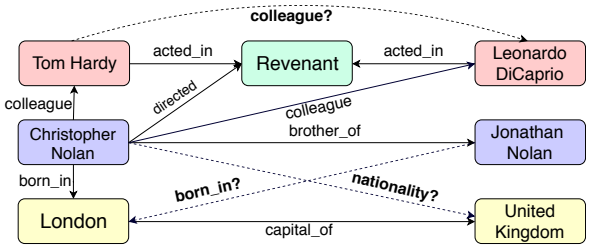


Figure 1: Subgraph of a knowledge graph contains actual relations between entities (solid lines) and inferred relations that are initially hidden (dashed lines).

In comparison, translational models like TransE (Bordes et al., 2013), DISTMULT (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) propose arguably simpler models. TransE considers the translation operation between head and tail entities for relations. DISTMULT (Yang et al., 2015) learns embeddings using a *bilinear diagonal model* which is a special case of the bilinear objective used in NTN and TransE. DISTMULT uses weighted element-wise dot products to model entity relations. ComplEx (Trouillon et al., 2016) generalizes DISTMULT (Yang et al., 2015) by using complex embeddings and Hermitian dot products instead. These translational models are faster, require fewer parameters and are relatively easier to train, but result in less expressive KG embeddings.

Recently, two CNN based models have been proposed for relation prediction, namely ConvE (Dettmers et al., 2018) and ConvKB (Nguyen et al., 2018). ConvE uses 2-D convolution over embeddings to predict links. It comprises of a convolutional layer, a fully connected projection layer and an inner product layer for the final predictions. Different feature maps are generated using multiple filters to extract global relationships. Concatenation of these feature maps represents an input triple. These models are parameter efficient but consider each triple independently without taking into account the relationships between the triples.

A graph based neural network model called R-GCN (Schlichtkrull et al., 2018) is an extension of applying *graph convolutional networks* (GCNs) (Kipf and Welling, 2017) to relational data. It applies a convolution operation to the neighborhood of each entity and assigns them equal weights. This graph based model does not outperform the CNN based models.

Existing methods either learn KG embeddings by solely focusing on entity features or by taking into account the features of entities and relations in a disjoint manner. Instead, our proposed graph attention model holistically captures multi-hop and semantically similar relations in the n-hop neighborhood of any given entity in the KG.

3 Our Approach

We begin this section by introducing the notations and definitions used in the rest of the paper, followed by a brief background on *graph attention networks* (GATs) (Veličković et al., 2018). Finally, we describe our proposed attention architecture for knowledge graphs followed by our decoder network.

3.1 Background

A knowledge graph is denoted by $\mathcal{G} = (\mathcal{E}, R)$, where $\mathcal{E}$ and R represent the set of entities (nodes) and relations (edges), respectively. A triple (e_s, r, e_o) is represented as an edge r between nodes e_s and e_r in $\mathcal{G}$[1]. Embedding models try to learn an effective representation of entities, relations, and a scoring function f, such that for a given input triple $t = (e_s, r, e_o)$, $f(t)$ gives the likelihood of t being a valid triple. For example, Figure 1 shows the subgraph from a KG which infers missing links represented by dashed lines using existing triples such as *(London, captial_of, United Kingdom)*.

3.2 Graph Attention Networks (GATs)

Graph convolutional networks (GCNs) (Kipf and Welling, 2017) gather information from the entity's neighborhood and all neighbors contribute equally in the information passing. To address the shortcomings of GCNs, (Veličković et al., 2018) introduced *graph attention networks* (GATs). GATs learn to assign varying levels of importance to nodes in every node's neighborhood, rather than treating all neighboring nodes with equal importance, as is done in GCN.

The input feature set of nodes to a layer is $\mathbf{x} = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_N\}$. A layer produces a transformed set of node feature vectors $\mathbf{x}' = \{\vec{x}'_1, \vec{x}'_2, ..., \vec{x}'_N\}$, where $\vec{x}_i$ and $\vec{x}'_i$ are input and output embeddings of the entity e_i, and N is number of entities (nodes). A single GAT layer can be described as

$$e_{ij} = a(\mathbf{W}\vec{x_i}, \mathbf{W}\vec{x_j}) \tag{1}$$

where e_{ij} is the attention value of the edge (e_i, e_j) in $\mathcal{G}$, $\mathbf{W}$ is a parametrized linear transformation matrix mapping the input features to a higher dimensional output feature space, and a is any *attention function* of our choosing.

Attention values for each edge are the *importance* of the edge $(e_i, e_j)'$s features for a source node e_i. Here, the relative attention α_{ij} is computed using a *softmax function* over all the values in the neighborhood. Equation 2 shows the output of a layer. GAT employs *multi-head attention* to stabilize the learning process as credited to (Vaswani et al., 2017).

$$\vec{x'_i} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{x_j}\right) \tag{2}$$

The multihead attention process of concatenating K attention heads is shown as follows in Equation 3.

$$\vec{x'_i} = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{x_j}\right) \tag{3}$$

where $\|$ represents concatenation, σ represents any non-linear function, α_{ij}^k are normalized attention coefficients of edge (e_i, e_j) calculated by the k-th attention mechanism, and $\mathbf{W}^k$ represents the corresponding linear transformation matrix of the k-th attention mechanism. The output embedding in the final layer is calculated using *averaging*, instead of the concatenation operation, to achieve multi-head attention, as is shown in the following Equation 4.

$$\vec{x'_i} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{x_j}\right) \tag{4}$$

3.3 Relations are Important

Despite the success of GATs, they are unsuitable for KGs as they ignore relation (edge) features,

[1] From here onwards, the pairs "*node / entity*" and "*edge / relation*" will be used interchangeably

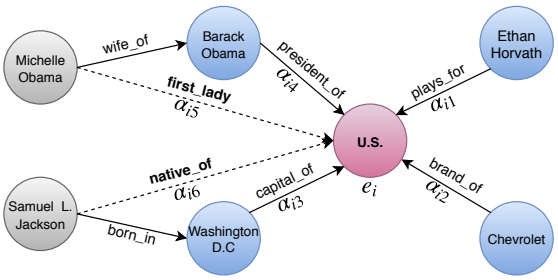


Figure 2: This figure shows the aggregation process of our graph attentional layer. α_{ij} represents relative attention values of the edge. The dashed lines represent an *auxiliary* edge from a n-hop neighbors, in this case $n = 2$.

which are an integral part of KGs. In KGs, entities play different roles depending on the *relation* they are associated with. For example, in Figure 1, entity *Christopher Nolan* appears in two different triples assuming the roles of a *brother* and a *director*. To this end, we propose a novel embedding approach to incorporate *relation and neighboring node features* in the attention mechanism. We define a single attentional layer, which is the building block of our model. Similar to GAT, our framework is agnostic to the particular choice of attention mechanism.

Each layer in our model takes two embedding matrices as input. *Entity embeddings* are represented by a matrix $\mathbf{H} \in \mathbb{R}^{N_e \times T}$, where the i-th row is the embedding of entity e_i, N_e is the total number of entities, and T is the feature dimension of each entity embedding. With a similar construction, the *relation embeddings* are represented by a matrix $\mathbf{G} \in \mathbb{R}^{N_r \times P}$. The layer then outputs the corresponding embedding matrices, $\mathbf{H}' \in \mathbb{R}^{N_e \times T'}$ and $\mathbf{G}' \in \mathbb{R}^{N_r \times P'}$.

In order to obtain the new embedding for an entity e_i, a representation of each triple associated with e_i is learned. We learn these embeddings by performing a linear transformation over the concatenation of entity and relation feature vectors corresponding to a particular triple $t_{ij}^k = (e_i, r_k, e_j)$, as is shown in Equation 5. This operation is also illustrated in the initial block of Figure 4.

$$\vec{c_{ijk}} = \mathbf{W}_1[\vec{h_i} \| \vec{h_j} \| \vec{g_k}] \tag{5}$$

where $\vec{c_{ijk}}$ is the vector representation of a triple t_{ij}^k. Vectors $\vec{h_i}, \vec{h_j}$, and $\vec{g_k}$ denote embeddings of entities e_i, e_j and relation r_k, respectively. Additionally, $\mathbf{W}_1$ denotes the linear transformation matrix. Similar to (Veličković et al., 2018), we learn

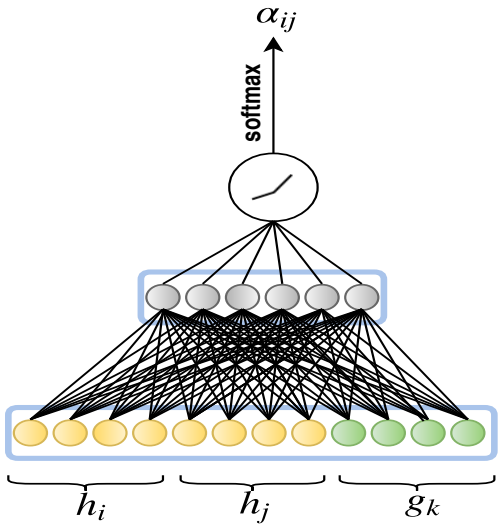


Figure 3: Attention Mechanism

the *importance* of each triple t_{ij}^k denoted by b_{ijk}. We perform a linear transformation parameterized by a weight matrix $\mathbf{W}_2$ followed by application of the LeakyRelu non-linearity to get the absolute attention value of the triple (Equation 6).

$$b_{ijk} = \text{LeakyReLU}\left(\mathbf{W}_2 c_{ijk}\right) \tag{6}$$

To get the relative attention values *softmax* is applied over b_{ijk} as shown in Equation 7. Figure 3 shows the computation of relative attention values α_{ijk} for a single triple.

$$\begin{aligned} \alpha_{ijk} &= \text{softmax}_{jk}(b_{ijk}) \\ &= \frac{\exp(b_{ijk})}{\sum_{n \in \mathcal{N}_i} \sum_{r \in \mathcal{R}_{in}} \exp(b_{inr})} \end{aligned} \tag{7}$$

where $\mathcal{N}_i$ denotes the neighborhood of entity e_i and $\mathcal{R}_{ij}$ denotes the set of relations connecting entities e_i and e_j. The new embedding of the entity e_i is the sum of each triple representation weighted by their attention values as shown in Equation 8.

$$\vec{h_i'} = \sigma\left(\sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{R}_{ij}} \alpha_{ijk} \vec{c_{ijk}}\right) \tag{8}$$

As suggested by (Veličković et al., 2018), multi-head attention which was first introduced by (Vaswani et al., 2017), is used to stabilize the learning process and encapsulate more information about the neighborhood. Essentially, M independent attention mechanisms calculate the embeddings, which are then concatenated, resulting in the following representation:

$$\vec{h_i'} = \Big\Vert_{m=1}^{M} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ijk}^m c_{ijk}^m\right) \tag{9}$$

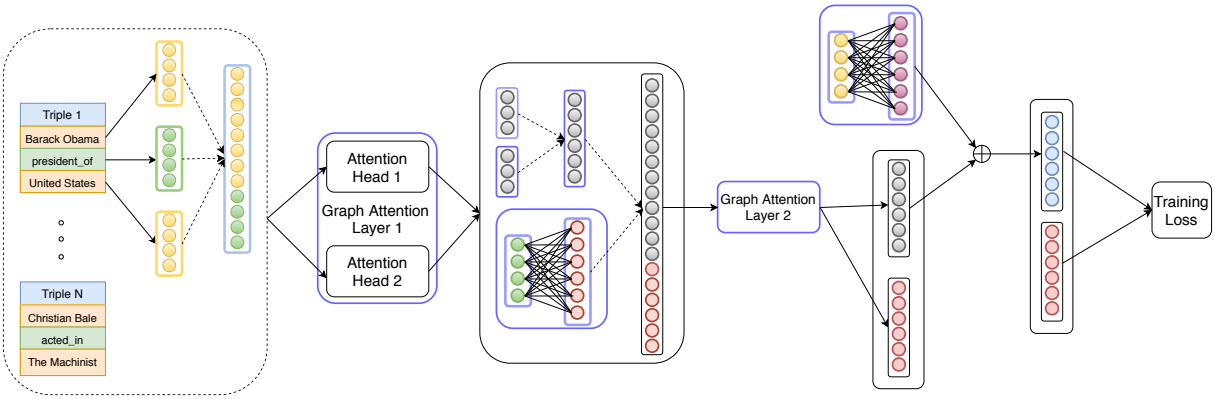


Figure 4: This figure shows end-to-end architecture of our model. Dashed arrows in the figure represent concatenation operation. Green circles represents initial entity embedding vectors and yellow circles represents initial relation embedding vectors.

This is the *graph attention layer* shown in Figure 4. We perform a linear transformation on input *relation embedding* matrix $\mathbf{G}$, parameterized by a weight matrix $\mathbf{W}^R \in \mathbb{R}^{T \times T'}$, where T' is the dimensionality of output *relation embeddings* (Equation 10).

$$G' = G.\mathbf{W}^R \tag{10}$$

In the final layer of our model, instead of concatenating the embeddings from multiple heads we employ averaging to get final embedding vectors for entities as shown in Equation 11.

$$\vec{h}_i' = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in\mathcal{N}_i}\sum_{k\in\mathcal{R}_{ij}}\alpha_{ijk}^m c_{ijk}^m\right) \tag{11}$$

However, while learning new embeddings, entities lose their initial embedding information. To resolve this issue, we linearly transform $\mathbf{H}^i$ to obtain $\mathbf{H}^t$ using a weight matrix $\mathbf{W}^E \in \mathbb{R}^{T^i \times T^f}$, where $\mathbf{H}^i$ represents the input entity embeddings to our model, $\mathbf{H}^t$ represents the transformed entity embeddings, T^i denotes the dimension of an initial entity embedding, and T^f denotes the dimension of the final entity embedding. We add this initial entity embedding information to the entity embeddings obtained from the final attentional layer, $\mathbf{H}^f \in \mathbb{R}^{N_e \times T^f}$ as shown in Equation 12.

$$\mathbf{H}'' = \mathbf{W}^E\mathbf{H}^t + \mathbf{H}^f \tag{12}$$

In our architecture, we extend the notion of an *edge* to a *directed path* by introducing an auxiliary relation for n-hop neighbors between two entities. The embedding of this auxiliary relation is the summation of embeddings of all the relations in the path. Our model iteratively accumulates knowledge from distant neighbors of an entity. As illustrated in figure 2, in the first layer of our model, all entities capture information from their *direct in-flowing neighbors*. In the second layer, *U.S* gathers information from entities *Barack Obama, Ethan Horvath, Chevrolet, and Washington D.C*, which already possess information about their neighbors *Michelle Obama* and *Samuel L. Jackson*, from a previous layer. In general, for a n layer model the incoming information is accumulated over a n-hop neighborhood. The aggregation process to learn new entity embeddings and the introduction of an auxiliary edge between n-hop neighbors is also shown in Figure 2. We normalize the entity embeddings after every generalized GAT layer and prior to the first layer, for every main iteration.

3.4 Training Objective

Our model borrows the idea of a *translational scoring function* from (Bordes et al., 2013), which learns embeddings such that for a given *valid* triple $t_{ij}^k = (e_i, r_k, e_j)$, the condition $\vec{h}_i + \vec{g}_k \approx \vec{h}_j$ holds, i.e., e_j is the nearest neighbor of e_i connected via relation r_k. Specifically, we try to learn *entity* and *relation* embeddings to minimize the L1-norm dissimilarity measure given by $d_{t_{ij}} = \|\vec{h}_i + \vec{g}_k - \vec{h}_j\|_1$.

We train our model using *hinge-loss* which is given by the following expression

$$L(\Omega) = \sum_{t_{ij}\in S}\sum_{t'_{ij}\in S'} \max\{d_{t'_{ij}} - d_{t_{ij}} + \gamma, 0\} \tag{13}$$

where $\gamma > 0$ is a margin hyper-parameter, S is the set of valid triples, and S' denotes the set of invalid

triples, given formally as

$$S' = \underbrace{\{t^k_{i'j} \mid e'_i \in \mathcal{E} \setminus e_i\}}_{\text{replace head entity}} \cup \underbrace{\{t^k_{ij'} \mid e'_j \in \mathcal{E} \setminus e_j\}}_{\text{replace tail entity}}$$

3.5 Decoder

Our model uses ConvKB (Nguyen et al., 2018) as a *decoder*. The aim of the convolutional layer is to analyze the global embedding properties of a triple t^k_{ij} across each dimension and to generalize the transitional characteristics in our model. The score function with multiple feature maps can be written formally as:

$$f(t^k_{ij}) = \left(\Big\|_{m=1}^{\Omega} \text{ReLU}([\vec{h}_i, \vec{g}_k, \vec{h}_j] * \omega^m) \right).\mathbf{W}$$

where ω^m represents the m^{th} convolutional filter, Ω is a hyper-parameter denoting number of filters used, $*$ is a convolution operator, and $\mathbf{W} \in \mathbb{R}^{\Omega k \times 1}$ represents a linear transformation matrix used to compute the final score of the triple. The model is trained using soft-margin loss as

$$\mathcal{L} = \sum_{t^k_{ij} \in \{S \cup S'\}} \log(1 + \exp(l_{t^k_{ij}} . f(t^k_{ij}))) + \frac{\lambda}{2} \|\mathbf{W}\|_2^2$$

$$\text{where } l_{t^k_{ij}} = \begin{cases} 1 & \text{for } t^k_{ij} \in S \\ -1 & \text{for } t^k_{ij} \in S' \end{cases}$$

4 Experiments and Results

4.1 Datasets

To evaluate our proposed method, we use five benchmark datasets: *WN18RR* (Dettmers et al., 2018), *FB15k-237* (Toutanova et al., 2015), *NELL-995* (Xiong et al., 2017), *Unified Medical Language Systems (UMLS)* (Kok and Domingos, 2007) and *Alyawarra Kinship* (Lin et al., 2018). Previous works (Toutanova et al., 2015; Dettmers et al., 2018) suggest that the task of relation prediction in *WN18* and *FB15K* suffers from the problem of *inverse relations*, whereby one can achieve state-of-the-art results using a simple *reversal rule* based model, as shown by (Dettmers et al., 2018). Therefore, corresponding subset datasets *WN18RR* and *FB15k-237* were created to resolve the reversible relation problem in *WN18* and *FB15K*. We used the data splits provided by (Nguyen et al., 2018). Table 1 provides statistics of all datasets used.

4.2 Training Protocol

We create two sets of invalid triples, each time replacing either the head or tail entity in a triple by an invalid entity. We randomly sample equal number of invalid triples from both the sets to ensure robust performance on detecting both head and tail entity. Entity and relation embeddings produced by TransE (Bordes et al., 2013; Nguyen et al., 2018) are used to initialize our embeddings.

We follow a two-step training procedure, i.e., we first train our generalized GAT to encode information about the graph entities and relations and then train a decoder model like ConvKB (Nguyen et al., 2018) to perform the relation prediction task. The original GAT update Equation 3 only aggregates information passed from 1-hop neighborhood, while our generalized GAT uses information from the n-hop neighborhood. We use auxiliary relations to aggregate more information about the neighborhood in sparse graphs. We use Adam to optimize all the parameters with initial learning rate set at 0.001. Both the entity and relation embeddings of the final layer are set to 200. The optimal hyper-parameters set for each dataset are mentioned in our supplementary section.

4.3 Evaluation Protocol

In the relation prediction task, the aim is to predict a triple (e_i, r_k, e_j) with e_i or e_j missing, i.e., predict e_i given (r_k, e_j) or predict e_j given (e_i, r_k). We generate a set of $(N-1)$ corrupt triples *for each entity* e_i by replacing it with every other entity $e_i' \in \mathcal{E} \setminus e_i$, then we assign a score to each such triple. Subsequently, we sort these scores in ascending order and get the rank of a correct triple (e_i, r_k, e_j). Similar to previous work ((Bordes et al., 2013), (Nguyen et al., 2018), (Dettmers et al., 2018)), we evaluate all the models in a *filtered* setting, i.e, during ranking we remove corrupt triples which are already present in one of the training, validation, or test sets. This whole process is repeated by replacing the tail entity e_j, and averaged metrics are reported. We report mean reciprocal rank (MRR), mean rank (MR) and the proportion of correct entities in the top N ranks (Hits@N) for $N = 1, 3$, and 10.

4.4 Results and Analysis

Tables 2 and 3 present the prediction results on the test sets of all the datasets. The results clearly demonstrate that our proposed method[2]

[2] Our work

Dataset	# Entities	# Relations	# Edges				Mean in-degree	Median in-degree
			Training	Validation	Test	Total		
WN18RR	40,943	11	86,835	3034	3134	93,003	2.12	1
FB15k-237	14,541	237	272,115	17,535	20,466	310,116	18.71	8
NELL-995	75,492	200	149,678	543	3992	154,213	1.98	0
Kinship	104	25	8544	1068	1074	10,686	82.15	82.5
UMLS	135	46	5216	652	661	6529	38.63	20

Table 1: Dataset statistics

	WN18RR					FB15K-237				
			Hits@N					Hits@N		
	MR	MRR	@1	@3	@10	MR	MRR	@1	@3	@10
DistMult (Yang et al., 2015)	7000	0.444	41.2	47	50.4	512	0.281	19.9	30.1	44.6
ComplEx (Trouillon et al., 2016)	7882	0.449	40.9	46.9	53	546	0.278	19.4	29.7	45
ConvE (Dettmers et al., 2018)	4464	**0.456**	**41.9**	47	53.1	245	0.312	22.5	34.1	49.7
TransE (Bordes et al., 2013)	2300	0.243	4.27	44.1	53.2	323	0.279	19.8	37.6	44.1
ConvKB (Nguyen et al., 2018)	**1295**	0.265	5.82	44.5	55.8	216	0.289	19.8	32.4	47.1
R-GCN (Schlichtkrull et al., 2018)	6700	0.123	8	13.7	20.7	600	0.164	10	18.1	30
Our work	1940	0.440	36.1	**48.3**	**58.1**	**210**	**0.518**	**46**	**54**	**62.6**

Table 2: Experimental results on WN18RR and FB15K-237 test sets. Hits@N values are in percentage. The best score is in **bold** and second best score is underlined.

	NELL-995					Kinship				
			Hits@N					Hits@N		
	MR	MRR	@1	@3	@10	MR	MRR	@1	@3	@10
DistMult (Yang et al., 2015)	4213	0.485	40.1	52.4	61	5.26	0.516	36.7	58.1	86.7
ComplEx (Trouillon et al., 2016)	4600	0.482	39.9	52.8	60.6	2.48	0.823	73.3	89.9	97.11
ConvE (Dettmers et al., 2018)	3560	0.491	40.3	53.1	61.3	2.03	0.833	73.8	91.7	**98.14**
TransE (Bordes et al., 2013)	2100	0.401	34.4	47.2	50.1	6.8	0.309	0.9	64.3	84.1
ConvKB (Nguyen et al., 2018)	**600**	0.43	37.0	47	54.5	3.3	0.614	43.62	75.5	95.3
R-GCN (Schlichtkrull et al., 2018)	7600	0.12	8.2	12.6	18.8	25.92	0.109	3	8.8	23.9
Our work	965	**0.530**	**44.7**	**56.4**	**69.5**	**1.94**	**0.904**	**85.9**	**94.1**	98

Table 3: Experimental results on NELL-995 and Kinship test sets. Hits@N values are in percentage. The best score is in **bold** and second best score is underlined.

significantly outperforms state-of-the-art results on five metrics for *FB15k-237*, and on two metrics for *WN18RR*. We downloaded publicly available source codes to reproduce results of the state-of-the-art methods[3][4][5][6][7][8] on all the datasets.

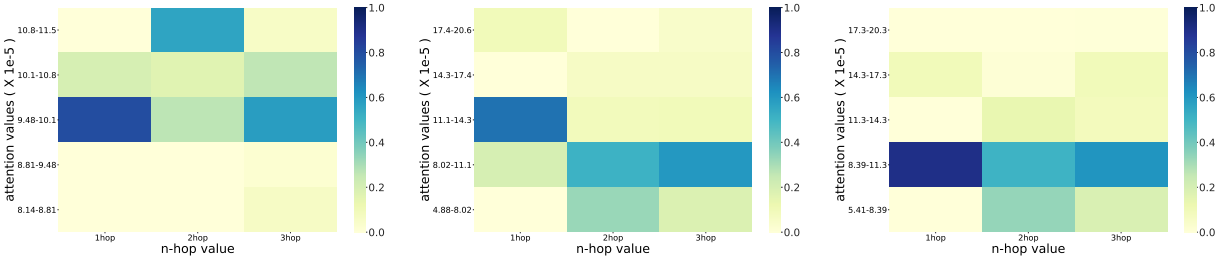
Attention Values vs Epochs: We study the distribution of attention with increasing epochs for a particular node. Figure 5 shows this distribution on *FB15k-237*. In the initial stages of the learning process, the attention is distributed randomly. As the training progresses and our model gathers more information from the neighborhood, it assigns more attention to direct neighbors and takes minor information from the more distant neighbors. Once the model converges, it learns to gather multi-hop and clustered relation information from the n-hop neighborhood of the node.

PageRank Analysis: We hypothesize that complex and hidden multi-hop relations among entities are captured more succinctly in dense graphs as opposed to sparse graphs. To test this hypothesis, we perform an analysis similar to ConvE, where they study the *correlation* between *mean PageRank* and *increase in MRR relative to DistMult*. We notice a strong correlation coefficient of $r = 0.808$. Table 4 indicates that when there is an increase in PageRank values, there is also a corresponding increase in MRR values. We observe an anomaly to our observed correlation in case of *NELL-995* versus *WN18RR* and attribute this to the highly sparse and *hierarchical structure* of *WN18RR* which poses as a challenge to our method that does not capture information in a top-down recursive fashion.

4.5 Ablation Study

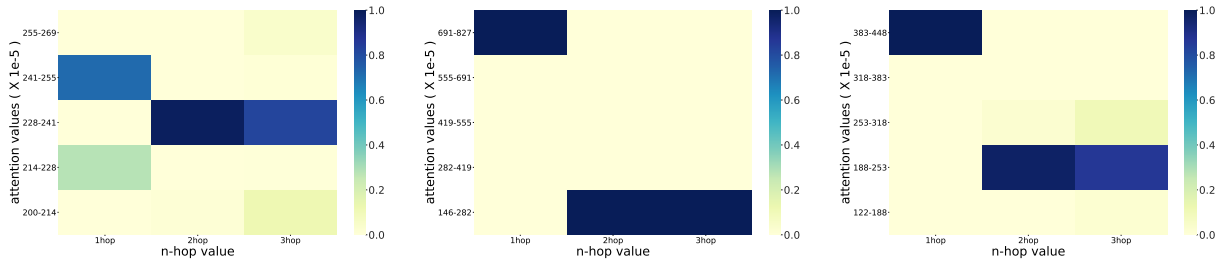
We carry out an ablation study on our model, where we analyze the behavior of *mean rank* on a test set when we omit *path generalization* ($-$PG), i.e., removing n-hop information, and omit *relation Information* ($-$Relations) from our model. Figure 7 shows that our model performs better

[3] TransE [4] DistMult [5] ComplEx [6] R-GCN [7] ConvE [8] ConvKB



(a) Epoch 0 (b) Epoch 1200 (c) Epoch 2400

Figure 5: Learning process of our model on FB15K-237 dataset. Y-axis represents attention values $\times 1e^{-5}$.



(a) Epoch 0 (b) Epoch 1800 (c) Epoch 3600

Figure 6: Learning process of our model on WN18RR dataset. Y-axis represents attention values $\times 1e^{-5}$

Dataset	PageRank	Relative Increase
NELL-995	1.32	0.025
WN18RR	2.44	-0.01
FB15k-237	6.87	0.237
UMLS	740	0.247
Kinship	961	0.388

Table 4: Mean PageRank $\times 10^{-5}$ vs relative increase in MRR wrt. DistMult.

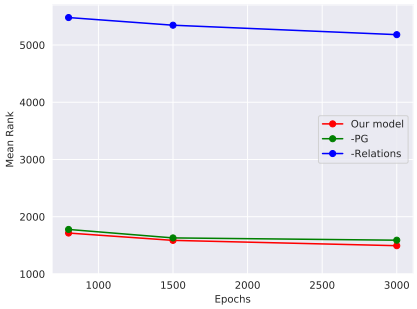


Figure 7: Epochs vs Mean Rank for our model and two ablated models on *NELL-995*. −PG (green) represents the model after removing n-hop auxiliary relations or *path generalization*, −Relations (blue) represents model without taking relations into account and Our model (red) represents the entire model.

than the two ablated models and we see a significant drop in the results when using ablated models on *NELL-995*. Removing the relations from the proposed model has a huge impact on the results which suggests that the relation embeddings play a pivotal role in relation prediction.

5 Conclusion and Future Work

In this paper, we propose a novel approach for relation prediction. Our approach improves over the state-of-the-art models by significant margins. Our proposed model learns new graph attention-based embeddings that specifically cater to relation prediction on KGs. Additionally, we generalize and extend graph attention mechanisms to capture both entity and relation features in a multi-hop neighborhood of a given entity. Our detailed and exhaustive empirical analysis gives more insight into our method's superiority for relation prediction on KGs. The proposed model can be extended to learn embeddings for various tasks using KGs such as dialogue generation (He et al., 2017; Keizer et al., 2017), and question answering (Zhang et al., 2016; Diefenbach et al., 2018).

In the future, we intend to extend our method to better perform on hierarchical graphs and capture higher-order relations between entities (like mo-

tifs) in our graph attention model.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. 2015. Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2013*, pages 1533–1544.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229, Uppsala, Sweden. Association for Computational Linguistics.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), 2014*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS), 2013*, pages 2787–2795.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), 2018*.

Dennis Diefenbach, Kamal Singh, and Pierre Maret. 2018. Wdaqua-core1: a question answering service for rdf knowledge bases. In *Companion of the The Web Conference 2018 on The Web Conference (WWW), 2018*, pages 1087–1091. International World Wide Web Conferences Steering Committee.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017*.

Simon Keizer, Markus Guhe, Heriberto Cuayahuitl, Ioannis Efstathiou, Klaus-Peter Engelbrecht, Mihai Dobre, Alex Lascarides, and Oliver Lemon. 2017. Evaluating persuasion strategies and deep reinforcement learning methods for negotiation dialogue agents. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (ACL), 2017*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR), 2017*.

Stanley Kok and Pedro Domingos. 2007. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, (ICML), 2007.

Lili Kotlerman, Ido Dagan, Bernardo Magnini, and Luisa Bentivogli. 2015. Textual entailment graphs. *Natural Language Engineering*, 21(5):699–724.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018*.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015*.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), 2018*, volume 2, pages 327–333.

Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. 2016. Holographic embeddings of knowledge graphs. In *(AAAI), 2016*, volume 2, pages 3–2.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, (ICML), 2011.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC), 2018*, pages 593–607.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013a. Reasoning with neural tensor networks for knowledge base completion. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in*

Neural Information Processing Systems 26 (NIPS), 2013, pages 926–934.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013b. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems (NIPS), 2013*, pages 926–934.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015*, pages 1499–1509.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML), 2016*, pages 2071–2080.

Jorge Carlos Valverde-Rebaza and Alneu de Andrade Lopes. 2012. Link prediction in complex networks based on cluster information. In *Proceedings of the 21st Brazilian Conference on Advances in Artificial Intelligence*, (SBIA), 2012, pages 92–101.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS), 2017*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations (ICLR), 2018*.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd International Conference on World Wide Web*, (WWW), 2014, pages 515–526.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *International Conference on Learning Representations (ICLR), 2015*.

Yuanzhe Zhang, Kang Liu, Shizhu He, Guoliang Ji, Zhanyi Liu, Hua Wu, and Jun Zhao. 2016. Question answering over knowledge base with neural attention combining global knowledge information. *arXiv preprint arXiv:1606.00979, 2016*.

A Results on UMLS dataset

We report results of our model on *UMLS* dataset in Table 5. *UMLS* is a relatively smaller knowledge graph with only 135 entities and 46 relations. Although this dataset is comparable to *Kinship* in size, it is relatively much sparser than *Kinship* with mean in-degree being 38.63, as opposed to 82.15 in *Kinship*. We show that despite the small size of the dataset, our model outperforms the baselines which indicates the robustness of our model.

B Indegree versus Attention

We analyze the attention flow of our encoder architecture for both the datasets. For each of these datasets we select two entities, one with higher in-degree (high-node) and the other one having lower in-degree (low-node). We visualize the attention values for n-hop neighbors of these entities, where $n = 1, 2, 3$.

In *FB15k-237*, the low-node has 10 direct neighbors and the high-node has 50 direct neighbors. For the low-node, the number of 2-hop neighbors is 439 and 3-hop neighbors is 1543. For the high-node, the number of 2-hop and 3-hop neighbors are 2939 and 6915, respectively. Figure 8a shows that a significant proportion of the 1-hop neighbors of our low-node have high attention values. However, this proportion increases as we move farther from the low-node. Figure 8b shows a reversal in this trend. For the high node, it can be seen that a significant proportion of entities in the 1-hop neighborhood get assigned higher attention values than 2 or 3-hop neighbors. As the low-node has just 10 direct neighbors, it tries to gather more information from its 2-hop and 3-hop neighbors, whereas in the case of the high-node most of the information is collected from its direct neighbors.

In *WN18RR*, the low-node has 5 direct neighbors and a high-node has 25 direct neighbors. For the low-node, the number of 2-hop neighbors is 19 and 3-hop neighbors is 57. For the high-node, the number of 2-hop and 3-hop neighbors are 67 and 84, respectively. Figure 9 shows that unlike in *FB15k-237*, the distribution of attention values is similar in the case of the low and high-node. Higher proportion of direct neighbors are being assigned higher attention values, and decreases with increase in distance from the node. Explanation of this behavior can be found in the inherent structure of the *WN18RR* dataset. *WN18RR* follows a strictly hierarchical structure which ensures that relatively more information is present at the first level rather than lower levels. Our model prioritizes the aggregation of information from the direct neighbors of the node, but at the same time makes use of the auxiliary information provided by auxiliary edges to learn the structure.

C Optimal Hyper-parameters

In this section we report the optimal set of hyper-parameters for both our attention model (table 6) and ConvKB (table 7). We use grid search over *Hits@10* to find these optimal parameters. We do not use batch training for our attention model, whereas we use a batch size of 128 for every dataset in our decoder model. Also, we use of a step learning rate scheduler which decays the learning rate by a factor of 0.5, after every 500 epochs in our attention model and after every 25 epochs in the decoder model. Some of the parameters reported in tables 6 and 7 are as follows: negative_ratio is the ratio of negative and positive triples in the training set, i.e., we sample negative_ratio negative triples per positive triple in the set. Margin corresponds to the value of γ in the hinge loss equation.

D N-hop Paths

Table 8 shows the number of n-hop paths existing in all datasets. We report unique paths between two entities $n = \{2, 3, 4, 5\}$, i.e., if there exist multiple paths between two entities of length n, we count it just once. For $n = 1$, we report all the paths present. Essentially this boils down to counting pairs of entities having a shortest path of length n between them.

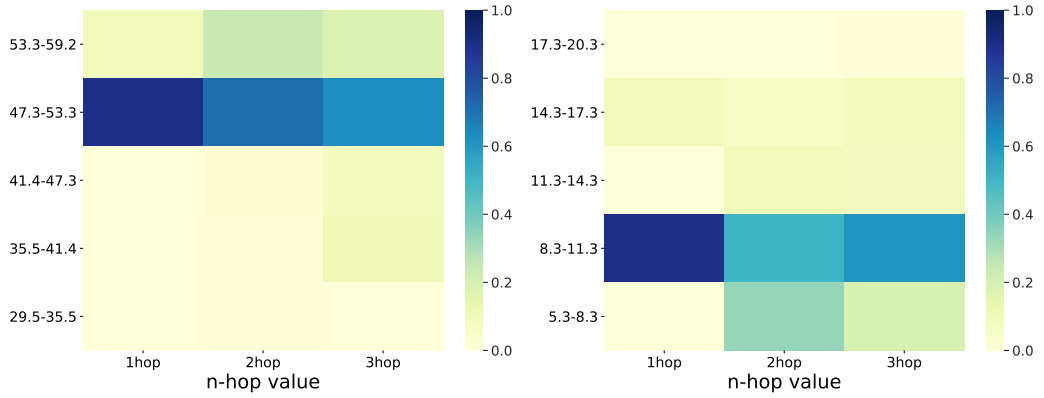
We make use of these paths to introduce auxiliary edges as discussed in the paper, so if a dataset has more n-paths, theoretically we can use the excess information to further improve the performance of our model. However, in datasets like *FB15k-237*, *Kinship* and *UMLS*, where the KG is already dense, the use of extra information can be safely neglected.

E Degree Distribution

We analyze the in-degree distributions for all datasets to get better insights. Figure 10 shows that *WN18RR* and *NELL-995* have significant number of nodes with no incoming information flow, i.e., nodes with zero in-degree in KG. We

	UMLS				
			Hits@N		
	MR	MRR	@1	@3	@10
DistMult (Yang et al., 2015)	512	0.281	19.9	30.1	44.6
ComplEx (Trouillon et al., 2016)	3.21	0.743	65.7	78.3	92.5
ConvE (Dettmers et al., 2018)	<u>1.38</u>	<u>0.935</u>	<u>89.8</u>	<u>96.7</u>	99
TransE (Bordes et al., 2013)	1.77	0.797	64.1	92.1	99.24
ConvKB (Nguyen et al., 2018)	1.66	0.785	60.8	96.14	<u>99.25</u>
R-GCN (Schlichtkrull et al., 2018)	24.9	0.204	12.6	20.8	32.5
KBGAT(this work)	**1.11**	**0.990**	**98.6**	**99.5**	**99.8**

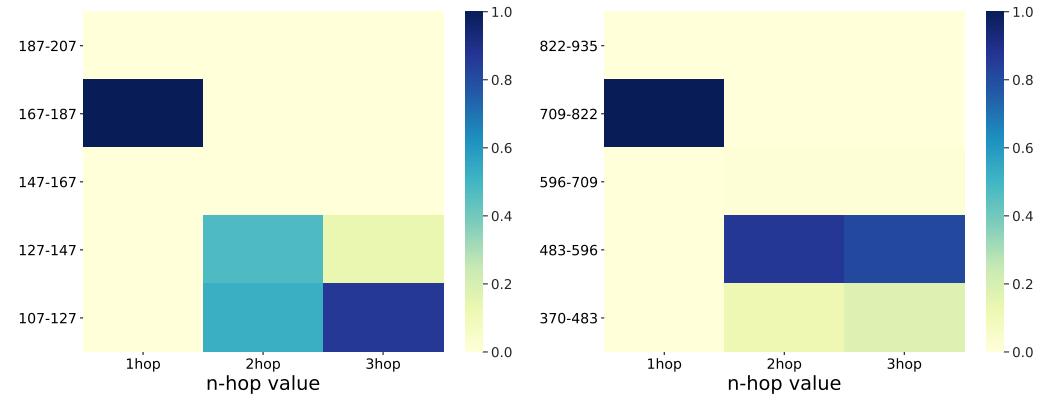
Table 5: Experimental results on UMLS test sets. Hits@N values are in percentage. The best score is in **bold** and second best score is <u>underlined</u>.



(a) Low indegree node

(b) High indegree node

Figure 8: This figure shows relation between n-hop paths of a node to its attention values in FB15k-237.



(a) Low indegree node

(b) High indegree node

Figure 9: This figure shows relation between n-hop paths of a node to its attention values in WN18RR.

	Weight decay	Epochs	negative_ratio	Learning Rate	Dropouts	Leaky Relu	nheads	Final dimensions	Margin
FB15k-237	$1e^{-5}$	3000	2	$1e^{-3}$	0.3	0.2	2	200	1
WN18RR	$5e^{-6}$	3600	2	$1e^{-3}$	0.3	0.2	2	200	5
NELL-995	$1e^{-5}$	3000	2	$1e^{-3}$	0.3	0.2	2	200	5
Kinship	$1e^{-5}$	3000	2	$1e^{-3}$	0.3	0.2	2	400	1
UMLS	$1e^{-5}$	3000	2	$1e^{-3}$	0.3	0.2	2	200	3

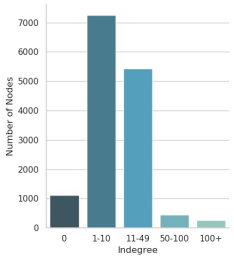
Table 6: Optimal values of hyperparameters for attention model are reported on all datasets.

	Weight decay	Epochs	negative_ratio	Learning Rate	Dropout	Filters
FB15k-237	$1e^{-5}$	200	40	$1e^{-3}$	0.3	50
WN18RR	$1e^{-5}$	200	40	$1e^{-3}$	0.0	500
NELL-995	$1e^{-5}$	200	40	$1e^{-3}$	0.0	500
Kinship	$1e^{-5}$	400	10	$1e^{-3}$	0.3	50
UMLS	$1e^{-5}$	400	10	$1e^{-3}$	0.3	50

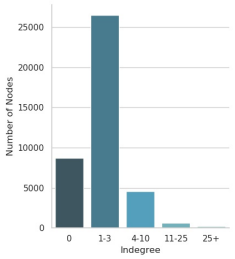
Table 7: Optimal values of hyperparameters for decoder(ConvKB) are reported on all datasets.

	1-hop	2-hop	3-hop	4-hop	5-hop
FB15k-237	272,115	12,792,938	46,019,137	60,756,091	31,825,944
WN18RR	86,835	207,376	450,748	979,522	2,100,993
NELL-995	149,678	3,736,186	21,561,431	67,455,032	99,591,081
Kinship	8544	2,168	0	0	0
UMLS	5216	6,941	4,572	976	190

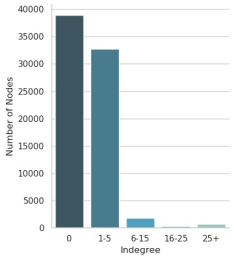
Table 8: This table shows number of n-hop paths in all datasets. We report unique paths for all values of n except $n = 1$.



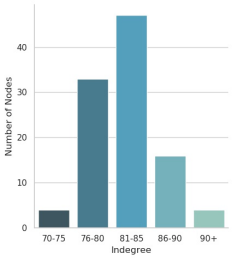
(a) FB15k-237



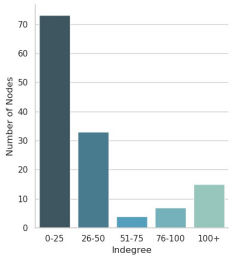
(b) WN18RR



(c) NELL-995



(d) Kinship



(e) UMLS

Figure 10: Indegree distribution for all datasets.

hypothesize that in *WN18RR*, the sparse and hierarchical nature makes it a relatively harder dataset. *FB15k-237* also has few rooted nodes in the KG, but since most of the nodes have many connections, the information flow is relatively effective in this case. In *Kinship* and *UMLS*, as we can see that there are no rooted nodes and both of these datasets are really dense in nature, especially *Kinship*. Due to this dense nature and the fact that every node learns from some other node, it is possible to predict relations effectively in these datasets.