electronics

MDPI

Article

Hierarchical Perceptual Graph Attention Network for Knowledge Graph Completion

Wenhao Han [1], Xuemei Liu [1,2,*], Jianhao Zhang [1] and Hairui Li [2]

[1] School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China; hanwenhao@stu.ncwu.edu.cn (W.H.); auaron1134@hotmail.com (J.Z.)

[2] School of Management and Economics, North China University of Water Resources and Electric Power, Zhengzhou 450046, China; lihairui@ncwu.edu.cn

* Correspondence: liuxuemei@ncwu.edu.cn

Abstract: Knowledge graph completion (KGC), the process of predicting missing knowledge through known triples, is a primary focus of research in the field of knowledge graphs. As an important graph representation technique in deep learning, graph neural networks (GNNs) perform well in knowledge graph completion, but most existing graph neural network-based knowledge graph completion methods tend to aggregate neighborhood information directly and individually, ignoring the rich hierarchical semantic structure of KGs. As a result, how to effectively deal with multi-level complex relations is still not well resolved. In this study, we present a hierarchical knowledge graph completion technique that combines both relation-level and entity-level attention and incorporates a weight matrix to enhance the significance of the embedded information under different semantic conditions. Furthermore, it updates neighborhood information to the central entity using a hierarchical aggregation approach. The proposed model enhances the capacity to capture hierarchical semantic feature information and is adaptable to various scoring functions as decoders, thus yielding robust results. We conducted experiments on a public benchmark dataset and compared it with several state-of-the-art models, and the experimental results indicate that our proposed model outperforms existing models in several aspects, proving its superior performance and validating the effectiveness of the model.

Keywords: knowledge graph completion; hierarchical semantic feature; graph neural network

Citation: Han, W.; Liu, X.; Zhang, J.; Li, H. Hierarchical Perceptual Graph Attention Network for Knowledge Graph Completion. *Electronics* **2024**, *13*, 721. https://doi.org/10.3390/electronics13040721

Academic Editors: Shangsong Liang and Zaiqiao Meng

Received: 10 January 2024
Revised: 6 February 2024
Accepted: 7 February 2024
Published: 9 February 2024

1. Introduction

In recent years, the exponential growth of information technology and data resources has generated significant interest in organizing and processing data effectively. Consequently, in 2012, Google introduced the concept of knowledge graphs, which have received widespread attention. In essence, a knowledge graph is a semantic network that stores structured knowledge as triples. Each triple is a fact pair consisting of (head entity, relation, tail entity) or (h, r, t). For example, (The Great Wall, IsLocatedIn, China).

Knowledge graphs have revolutionized many solution paradigms in natural language processing and bolstered numerous downstream applications of artificial intelligence. Representative examples include recommender systems [1], question answering [2], and dialogue systems [3]. Although knowledge graphs such as FreeBase [4], YAGO [5], and WordNet [6] have incorporated millions of triples, they are still not enough to meet demand, as modern society continues to evolve and knowledge expands dramatically [7]. Consequently, this prompts us to undertake the task of predicting missing links, termed knowledge graph completion or link prediction.

One prevalent technique for complementing knowledge graphs is via knowledge graph embedding. Typically, this technique involves utilizing the existing fact triples in the knowledge graph as a foundation, and embedding entities and relations into low-dimensional vectors to obtain their knowledge representation. Ultimately, the trustworthiness of each fact triple is assessed by optimizing a scoring function. Knowledge graph embedding models can be classified into three categories: translation distance models [8–11], bilinear models [12–15], and neural network models [16–19]. While these models have achieved impressive performance, they neglect significant semantic information, such as graph structure information. As a result, graph neural networks (GNNs) have emerged. GNN-based models are capable of effectively capturing graph information and propagating it hierarchically through various graph aggregation mechanisms [20,21] to obtain the corresponding entity embeddings. For instance, CompGCN [22] suggests constructing an encoder–decoder framework in knowledge graph complementation utilizing the exceptional aggregation capability of a GCN as an encoder, and then utilizing a convolutional neural network (CNN) as a decoder for scoring purposes.

Although current graph neural networks have made considerable progress in aggregating graph information, they still lack the ability to extract diverse semantic hierarchies of graphs. An example of a graph for *Harry Potter* and its associated characters can be seen in Figure 1. The entity *Harry Potter* has four distinct relations: *friend_of*, *enemy_of*, *teacher_of*, and *married*. When the entity *Harry Potter* is paired with a particular relation, the effect varies depending on the different semantic characteristics of the relation. Different relations hold various levels of importance for the central entity, as seen in this example mapping where the *friend_of* relation is stronger for *Harry Potter* than the *married* relation. Moreover, distinct entities within the same relation also hold different levels of significance for the central entity. Different individuals who are connected to *Harry Potter* through the *friend_of* relation are {*Ron Weasley*, *Hermione Granger*, *Neville Longbottom*}. However, our focus will be primarily on the first two protagonists. Furthermore, it should be observed that the significance of an entity varies in various triple compositions. For instance, Ron Weasley has a different semantic importance when he is linked to *Hermione Granger* as a tail entity in a *married* relation than when he is linked to *Harry Potter* as a head entity in a *friend_of* relation.

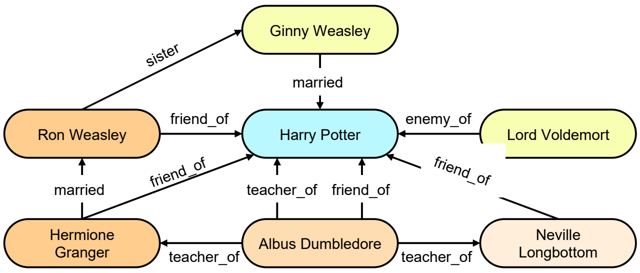

Figure 1. An illustrative example of a knowledge graph. Centre entity connects different tail entities through different relations.

In this paper, we present a novel Hierarchical Perceptual Graph Attention Network (HPGAT) that uses hierarchical attention to aggregate information from neighborhood entities and relation features. Initially, the attention mechanism [23,24] at the relation level combines the features of the central entity and the relations to create entity–relation vector embedding. Subsequently, the attention coefficients for each relation are acquired through the attention mechanism. For each entity involved in a particular relation under the central entity, the entity-level attention mechanism combines the features of the relation and different entities, creating relation–entity vector embedding. Then, entity attention is calculated, and the attention coefficients for each triple level can be obtained. Finally, the vector embedding of the central entity is updated by aggregating the feature information from each neighborhood triple in a hierarchical manner.

The contributions of our work are summarized as follows:

- We propose HPGAT, which is based on the attention mechanism that aggregates information by learning the hierarchical structure information in a given knowledge graph and the importance of entities and relations in different semantics.
- We implement HPGAT, which can hierarchically aggregate neighborhood feature information through entity-level attention and relation-level attention and obtain semantic weights of entities under different triples through weight matrices to obtain more accurate central entity embeddings.
- We conducted a number of comparison and ablation experiments on different datasets to validate the effectiveness of our model. The experimental results show that our model HPGAT outperforms the state-of-the-art models in knowledge graph complementation, demonstrating the effectiveness of the hierarchical structure for knowledge graph complementation.

2. Related Work

2.1. Knowledge Graph Embedding

The principal objective of embedding knowledge graphs is to master the technique of representing entities and relationships within a low-dimensional, distributed framework [25]. Broad classifications of these models include those based on translational distances, those that employ semantic matching techniques, and those that utilize neural network architectures. For every triple (h, r, t), the translation distance model interprets the relationship r as a transformation that maps the head entity h to the tail entity t. TransE [8], an initial portrayal of such models, roughly depicts each triple by $h + r \approx t$. However, TransE [8] exhibits limitations in its capacity to handle intricate relational patterns, including one-to-many, many-to-one, and many-to-many associations. Consequently, several derived models such as TransH [9], TransD [10], and TransR [11], among others, attempt to project the representations of entities and relations onto alternative spaces in order to address the challenge of representing complex relations.

Semantic models evaluate the likelihood of a fact represented as a triple by employing a scoring function based on similarity, which aligns the potential meanings of both entities and relationships within the vector space. The bilinear model known as RESCAL [12] pioneered this approach using tensors and matrices to encapsulate entities and relationships, respectively. Subsequent advancements based on RESCAL's framework have led to the development of models like ComplEx [15], Distmult [13], and HolE [14].

Neural network models consist of neural tensor networks (NTNs) [19] and convolutional neural network-based models. Among these, NTN [19] maps entities onto an input layer and integrates the embeddings of both head and tail entities by employing a distinct relational tensor. This tensor is utilized as input to calculate the score of the nonlinear layer. Convolutional neural networks offer efficient parameters and fast training, making them widely used in KGE. Convolutional embedding (ConvE) [16] produced outstanding outcomes via feature filters on reshaped feature matrices and relational embeddings. Convolutional knowledge base embedding (ConvKB) [18] enhances contemporary models by its ability to encapsulate global relationships and knowledge within its framework. Meanwhile, InteractE [26] augments the efficacy of ConvE by employing feature alignment, square reshaping, and the use of circular convolution to refine its performance. DBKGE [27] works by dynamically tracking the semantic representation of entities over time in a joint metric space and making predictions into the future.

2.2. Graph Neural Network Model

To overcome the constraints of traditional neural network structures (e.g., CCN) that can exclusively handle Euclidean data, researchers have developed graph convolutional neural networks (GCNs) [20]. These networks assign identical weight to each entity and carry out convolutional operations on its neighborhood. In contrast, R-GCNs [28] employ specific relational transformations during neighborhood aggregation, thereby demonstrating their efficacy in link prediction and entity classification. KBGAT [29] learns GAT-based embedding and introduces relational features, enabling it to capture richer multi-hop neighborhood feature information. In contrast, CompGCN [22] possesses a generic framework and uses a composition-based GCN as an encoder and ConvE [16] as a decoder, which allows for the simultaneous embedding of both entities and relations in KG. However, the current models concentrate solely on the feature information of entities and relations and fail to consider the elaborate hierarchical structure of the graph. This, in turn, results in them being unable to efficiently adapt to the hierarchical semantics between entities and relations during feature embedding. In this paper, we propose a GNN framework that utilizes the hierarchical attention mechanism. Our proposed framework and experimental outcomes are discussed in Section 3 and Section 4, respectively.

3. The Proposed HPGAT

In this section, we present a comprehensive description of HPGAT. The overall architecture is depicted in Figure 2. HPGAT comprises three components: (1) relation-level attention; (2) entity-level attention; and (3) hierarchical-based information aggregation. The first step in relation-level attention involves combining the features of entities and relations. Entity-level attention involves the partial aggregation of semantic information via entity paths that are connected to relations. Finally, hierarchical aggregation integrates entity-level attention and relation-level attention for propagating features.

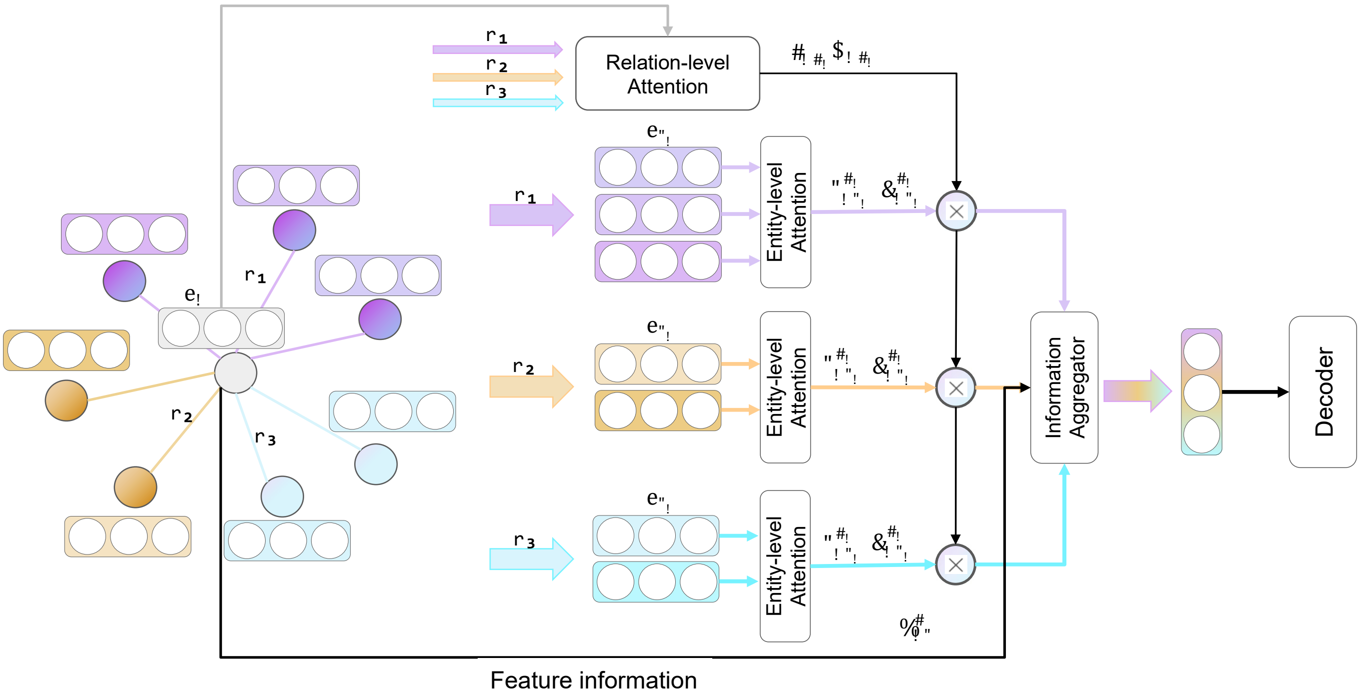

Figure 2. The overall structure of HPGAT. The HPGAT model is structured around three core modules: (1) relation-level attention, (2) entity-level attention, (3) hierarchical-based information aggregation. Firstly, the corresponding attention scores are calculated by relation-level attention and entity-level attention; then, the vector representation of the central entity is updated by information aggregation, and finally it is fed into the decoder.

3.1. Relation-Level Attention

In a knowledge graph, a particular entity's neighborhood structure can comprise one or more relations, and the significance of various relations in representing that entity varies considerably. Thus, the aggregation of each entity's neighborhood relation features directly is unsuitable. Consequently, we propose leveraging the attention mechanism for merging the semantic properties of assorted relations with the semantic data of the entity to derive its attention coefficient. In our framework, a knowledge graph is represented as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ denotes relations, and $\mathcal{T}$ constitutes the set of edges connecting these entities. Each edge, denoted as (h, r, t), signifies the existence of a relation $r \in \mathcal{R}$ from entity h to t. Following previous works [22,28], we extend $\mathcal{T}$ and $\mathcal{R}$ with homologous self-referential and inverse relations.

At this point, the edges and relations are extended as follows:

$$\mathcal{T}' = \mathcal{T} \cup \{(t, r^{-1}, h) \big| (h, r, t) \in \{\mathcal{T}\}\} \cup \{(h, \perp, h) | h \in \mathcal{E}\}, \mathcal{R}' = \mathcal{R} \cup \mathcal{R}_{inv} \cup \{\perp\}$$

where $\mathcal{R}_{inv} = \{r^{-1} | r \in \mathcal{R}\}$ and $\perp$ denote the inverse and self-loop relations, respectively.

To aggregate feature information for entity–neighborhood relations and distinguish between different weights for the same entity acting as head and tail, it is intuitive to learn a separate weight matrix for each entity. Additionally, we propose using special weights for relations to extract relation-specific features.

HPGAT obtains the embedding $\boldsymbol{m}_{hr}$ of the entities and relations by splicing them and subsequently feeds it as input to the attention layer.

$$\boldsymbol{m}_{hr} = W_1[W_h \boldsymbol{e}_h || W_r \boldsymbol{r}] \tag{1}$$

where $||$ denotes the concatenation operation, and $\boldsymbol{e}_h$ and $\boldsymbol{r}$ are the embeddings of entities and relations, respectively. $W_h \in \mathbb{R}^{d_0}$ and $W_r \in \mathbb{R}^{d_0}$ represent the trainable weight matrices of entities and relations, respectively, and $W_1 \in \mathbb{R}^{d_0 \times 2d_0}$ signifies the linear transformation matrix which maps the embedding vectors of entities–relations into a vector space to facilitate the learning of embedded features effectively.

At the attention layer, activation values of the embedding vectors are obtained through a linear transformation matrix. The LeakyReLU nonlinear activation function is then applied to acquire the attention value score.

$$\boldsymbol{s}_{hr} = LeakyReLu(W_2 \boldsymbol{m}_{hr}) \tag{2}$$

To ensure the comparability of attention values, we apply the SoftMax function to the attention value, resulting in the attention coefficient.

$$\alpha_{hr} = softmax_r(\boldsymbol{s}_{hr}) = \frac{exp(\boldsymbol{s}_{hr})}{\sum_{r' \in \mathcal{N}_h} exp(\boldsymbol{s}_{hr'})} \tag{3}$$

where $\mathcal{N}_h$ denotes the neighborhood of entity h.

3.2. Entity-Level Attention

After aggregating the relations, we notice that when an entity connects different entities through a specific relation, the importance of each entity to the central entity may be different, which leads to the fact that the process of aggregating the information by considering all the entities as having the same importance cannot effectively extract the hierarchical relations in the semantic structure. Therefore, to obtain more complete hierarchical semantic information, it is necessary to distinguish the importance characteristics of different entities. To solve these problems, we propose an entity-level attention mechanism and apply the same weight matrix to distinguish the semantic features of head and tail entities.

We partition the entities for the different entities under the action of a given relation and splice the relation with the entities under the action of the relation to obtain the feature embedding $\boldsymbol{c}_{ht}^{r}$.

$$\boldsymbol{c}_{ht}^{r} = W_3[W_r \boldsymbol{r} || W_t \boldsymbol{e}_t] \tag{4}$$

where $W_3 \in \mathbb{R}^{d_0 \times 2d_0}$ is the linear transformation matrix and W_r and $W_t \in \mathbb{R}^{d_0}$ are the trainable weight matrices for relations and entities, respectively. $\boldsymbol{e}_t$ is the embedding of entities.

At the entity-level attention layer, we similarly use a nonlinear activation function to obtain an attention score.

$$\beta_{ht}^{r} = softmax(\boldsymbol{c}_{ht}^{r}) = \frac{exp(LeakyReLu(\boldsymbol{c}_{ht}^{r}))}{\sum_{t' \in \mathcal{N}_{h,r}} exp(LeakyReLu(\boldsymbol{c}_{ht'}^{r}))} \tag{5}$$

where $\mathcal{N}_{h,r}$ denotes the tail entity of entity h under relation r.

While hierarchical attention is effective in extracting hierarchical information from the graph structure, it is crucial to avoid the model paying excessive attention to such features and neglecting the data's feature information during the training process. As a solution, we introduce a message module that transmits all feature information, mitigating the issue of weakened feature information due to the model's over-attention to hierarchical structures.

$$\boldsymbol{g}_{ht}^{r} = W_4[W_h\boldsymbol{e}_h||W_r\boldsymbol{r}||W_t\boldsymbol{e}_t] \tag{6}$$

where $W_4 \in \mathbb{R}^{d_0 \times 3d_0}$ is the linear transformation matrix.

3.3. Hierarchical-Based Information Aggregation

Updated embedded representations are obtained through information aggregation, which involves aggregating local information to central entities. In hierarchical structures, for the purpose of obtaining updated entity embeddings, it is preferred to perform an aggregation of entities and relations in a stepwise manner.

$$\boldsymbol{e}_h' = \sigma\left(\sum_{r \in \mathcal{N}_h} \sum_{t \in \mathcal{N}_{h,r}} \alpha_{hr}\boldsymbol{s}_{hr}\beta_{ht}^{r}\boldsymbol{c}_{ht}^{r} + \boldsymbol{g}_{ht}^{r}\right) \tag{7}$$

Multi-head attention is suggested in [30] to stabilize the learning process and enhance performance. In our study, we utilize multi-head attention to facilitate the model in capturing semantic features from various levels of the relational parameter space, resulting in an improved model fit. We transition the tandem operation to an averaging operation for reduced computational complexity. Consequently, the embedding of the final message is calculated as follows:

$$\boldsymbol{e}_h' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K} \sum_{r \in \mathcal{N}_h} \sum_{t \in \mathcal{N}_{h,r}} \alpha_{hr}^{k}\boldsymbol{m}_{hr}^{k}\beta_{ht}^{rk}\boldsymbol{c}_{ht}^{rk} + \boldsymbol{g}_{ht}^{rk}\right) \tag{8}$$

3.4. Decoder

In our work, we used TransE, DistMult and ConvE as decoders. Among these, ConvE exhibited superior performance. ConvE captures complex interactions between entities and relations by using convolutional neural networks. When processing a given knowledge graph triple (h, r, t), ConvE first reshapes the embedding vectors of the head entities and relations to form a two-dimensional tensor, and then performs a standard convolutional operation on the reshaped tensor to compute the score of the knowledge triples. In ConvE [16], the scores of knowledge triples are:

$$f(h, r, t) = ReLU\big(vec\big(ReLU([\boldsymbol{e}_h; \boldsymbol{r}] * \omega)\big)W\big)\boldsymbol{e}_t \tag{9}$$

where $\boldsymbol{e}_h$ and $\boldsymbol{r}$ are 2D reshapings of h and r. ω denotes a set of filters and $*$ denotes the convolution operator. $vec(\cdot)$ is a vectorization function, and W is the weight matrix. To train the model, standard cross entropy loss with label smoothing is optimized.

$$\mathcal{L} = -\frac{1}{N}\sum_{i}\big(t_i \cdot log(p_i) + (1 - t_i) \cdot log(1 - p_i)\big) \tag{10}$$

where t_i is the label of triple i and p_i is the corresponding score.

4. Experiments

To assess the efficacy of our proposed model, we carried out numerous experiments and presented comprehensive analysis results. Following this, we assessed the model's ability to predict links, comparing it against the baseline model, and confirmed its validity.

4.1. Experimental Setup

4.1.1. Dataset

Our model is evaluated for validation on two open-source datasets: WN18RR [16] and FB15K-237 [30], specifically. One disadvantage of WN18 [8] and FB15K [8] is test set leakage, which the WN18RR and FB15K-237 datasets addressed by eliminating inverse relations. The WN18RR dataset consists of 41K entities and 11 relations from WordNet. On the other hand, FB15K-237 contains 15K entities and 237 relations from Freebase. Detailed information about the two datasets can be found in Table 1.

Table 1. Dataset statistics of FB15k-237 and WN18RR.

Dataset	#Entities	#Relations	#Training	#Validation	#Test
WN18RR	40,943	11	86,835	3034	3134
FB15k-237	14,541	237	272,115	17,535	20,466

4.1.2. Evaluation Metrics

We use several evaluation metrics to assess the model effectiveness, among which are mean rank (MR), mean reciprocal rank (MRR), and $Hits@n$ (for n = 1, 3, and 10), respectively. In addition, we predicted the head entity by incorporating the inverse relation [31].

The formulas for MRR and MR are shown below, respectively.

$$MRR = \frac{1}{|S|}\sum_{i=1}^{|s|}\frac{1}{rank_i} = \frac{1}{|S|}\left(\frac{1}{rank_1} + \frac{1}{rank_2} + \cdots + \frac{1}{rank_{|s|}}\right) \tag{11}$$

$$MR = \frac{1}{|S|}\sum_{i=1}^{|s|} rank_i = \frac{1}{|S|}\left(rank_1 + rank_2 + \cdots + rank_{|s|}\right) \tag{12}$$

where S is the set of triples, $|S|$ is the number of triple sets, and $rank_i$ is the link prediction rank of the i_{th} triple. For MRR, a larger value corresponds to a better modeling effect, and the opposite is true for MR.

$Hits@n$ is the average proportion of triples with rank less than n in the link prediction. This is derived by

$$Hits@n = \frac{1}{|S|}\sum_{i=1}^{|s|} \mathbb{I}\,(rank_i \leq n) \tag{13}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The value of the function is 1 if the condition is true and 0 if the condition is false. A larger value corresponds to a better modeling effect.

4.1.3. Comparison Models

We compared our proposed model with many existing models so as to derive the validity and excellence of our model in a comparative test comprising translational distance models (TransE [8], RotatE [32], MuRP [33] and PairRE [34]), semantic models (DistMult [13] and ComplEx [15]), neural network models (ConvE [16], ConvKB [18], DeepER [35] and InteactE [26]), and GCN-based models (R-GCN [28], MRGAT [36] and CompGCN [29]).

4.1.4. Parameter Settings

We implemented the entire model in Pytorch (https://pytorch.org/). For the optimizer of the model, we used Adam to obtain the best results. For the hyperparameters of the model, we obtained them through grid search, and the hyperparameters that gave us better results were as follows: learning rate was 0.001, label smoothing was 0.1, and input and output dimensions were 100 and 200, respectively. On the FB15k-237 dataset, we used two attention headers with a batch size of 2048, whereas on WN18RR we used one attention header with a batch size of 256. We initialized the model parameters with Xavier.

4.2. Performance of HPGAT

This section provides a summary of how HPGAT compares to the baseline model, and Table 2 presents the MRR and $Hits@n$ results on the FB15k-237 and WN18RR datasets. The baseline model scores were obtained from previous papers [22,26,32] and the respective source papers of the models. The best results are in bold, and the second-best results are underlined. HPGAT improves the MRR on FB15k-237 by about 3% over CompGCN and by about 2% over $Hit@10$, showing the effect of leveraging the hierarchical structure of the knowledge graph. Compared to the other baselines, HPGAT outperforms all other methods in all five metrics on FB15k-237 and in four out of five metrics on WN18RR. Our study shows that our proposed HPGAT model outperforms existing link prediction models, demonstrating its effectiveness.

Table 2. Link prediction results of HPGAT on FB15k-237 and WN18RR. The best results are in bold and the second-best results are underlined.

Models	**FB15k-237**				**WN18RR**			
	MRR	**Hits**			**MRR**	**Hits**		
		@1	**@3**	**@10**		**@1**	**@3**	**@10**
TransE	0.257	0.174	0.284	0.420	0.182	0.027	0.295	0.444
DistMult	0.241	0.155	0.263	0.419	0.430	0.390	0.440	0.490
ComplEx	0.247	0.158	0.275	0.428	0.440	0.410	0.460	0.510
RotatE	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571
ConvE	0.325	0.237	0.356	0.501	0.430	0.400	0.440	0.520
ConvKB	0.243	0.155	0.371	0.421	0.249	0.057	0.417	0.524
R-GCN	0.249	0.151	0.264	0.417	0.123	0.080	0.137	0.207
MuRP	0.335	0.243	0.367	0.518	0.481	0.440	0.495	0.566
PairRE	0.351	0.256	0.387	0.544	-	-	-	-
MRGAT	0.355	0.266	0.392	0.539	0.481	**0.449**	0.495	0.544
DeepER	0.345	0.255	0.379	0.525	0.476	0.446	0.490	0.535
InteractE	0.354	0.263	0.386	0.535	0.463	0.430	0.483	0.528
CompGCN	0.355	0.264	0.390	0.535	0.479	0.443	0.494	0.546
HPGAT(ours)	**0.365**	**0.276**	**0.398**	**0.545**	**0.485**	0.445	**0.497**	**0.567**

4.3. Evaluation on Different Relation Categories

For intricate relations, our investigation centers on 1-N, N-1, and N-N relations (as illustrated in Table 3). As FB15k-237 possesses a more abundant variety of relation types and a denser graph structure, it was selected for comparison with InteractE [26], RotatE [32], and COMPGCN [22]. Our findings demonstrate that HPGAT surpasses the baseline model in most relation types. We have observed that RotatE performs better in simple 1-1 relations, presumably due to its capacity to capture diverse relational patterns like symmetry/asymmetry, inversion, and combination. In contrast, our preference is to capture the complex hierarchical relations within the graph and extract the rich underlying associations in entities and relations. This accounts for the superiority of our modelling results compared to others.

Table 3. Results of link prediction by relation category on FB15k-237 dataset. Following [9], relations were classified into four categories: one-to-one (1-1), one-to-many (1-N), many-to-one (N-1), and many-to-many (N-N). The best results are in bold.

		InteractE		**RotatE**		**COMPGCN**		**HPGAT**	
		MRR	**H@10**	**MRR**	**H@10**	**MRR**	**H@10**	**MRR**	**H@10**
Head Pred	1-1	0.386	0.547	**0.498**	0.593	0.457	0.604	0.461	**0.598**
	1-N	0.106	0.192	0.092	0.174	**0.112**	0.190	0.110	**0.213**
	N-1	0.466	0.647	0.471	0.674	0.471	0.656	**0.472**	**0.659**
	N-N	0.276	0.476	0.261	0.476	0.275	0.474	**0.278**	**0.483**
Tail Pred	1-1	0.368	0.547	**0.484**	0.578	0.453	0.589	0.457	**0.625**
	1-N	0.777	0.881	0.749	0.674	0.779	0.885	**0.789**	**0.892**
	N-1	0.074	0.141	0.074	0.138	0.076	**0.151**	**0.077**	0.146
	N-N	0.395	0.617	0.364	0.608	0.395	0.616	**0.400**	**0.621**

4.4. Evaluation on Different Decoders

Working with various decoders can enhance the robustness of a model. Therefore, we implemented several decoders: TransE, DistMult, and ConvE. The statistical results are given in Table 4. We evaluated the impact of different scoring functions on the model, and among all the decoders, ConvE yielded the best outcomes. We conclude that when used in conjunction with graph convolutional networks, ConvE can extract graphical structure information, resulting in an improved model performance.

Table 4. Performance of link prediction task evaluated on FB15k-237 dataset. Similar to COMPGCN, X + M (Y) denotes that method M is used for obtaining entity (and relation) embeddings with X as the scoring function. Y denotes the composition operator used. The best results are in bold.

Scoring Function(X)	**TransE**			**DistMult**			**ConvE**		
Methods	**MRR**	**MR**	**Hit@10**	**MRR**	**MR**	**Hit@10**	**MRR**	**MR**	**Hit@10**
X	0.294	357	0.465	0.241	354	0.419	0.325	244	0.501
X+D-GCN	0.299	351	0.469	0.321	255	0.497	0.344	200	0.524
X+W-GCN	0.264	1520	0.444	0.324	229	0.504	0.244	201	0.525
X+COMPGCN(sub)	0.335	194	0.514	0.336	231	0.513	0.352	199	0.530
X+COMPGCN(Mult)	0.337	233	0.515	0.338	**200**	**0.518**	0.353	216	0.532
X+COMPGCN(Corr)	0.336	214	0.518	0.335	227	0.514	0.355	**197**	0.535
X+HPGAT(ours)	**0.341**	**185**	**0.522**	**0.339**	230	0.516	**0.365**	216	**0.545**

4.5. Multi-Head Attention Mechanism

We used multi-head attention in our model to stabilize the learning process and improve performance. To explore the effect of different numbers of attention heads on the results, we used one, two and three attention heads for comparison experiments, and the results are shown in Figure 3. The results show that two attention heads are recommended for optimal performance with the FB15k-237 dataset. With the WN18RR dataset, the more attention heads the worse the result is, and when there is one, the best result is achieved.

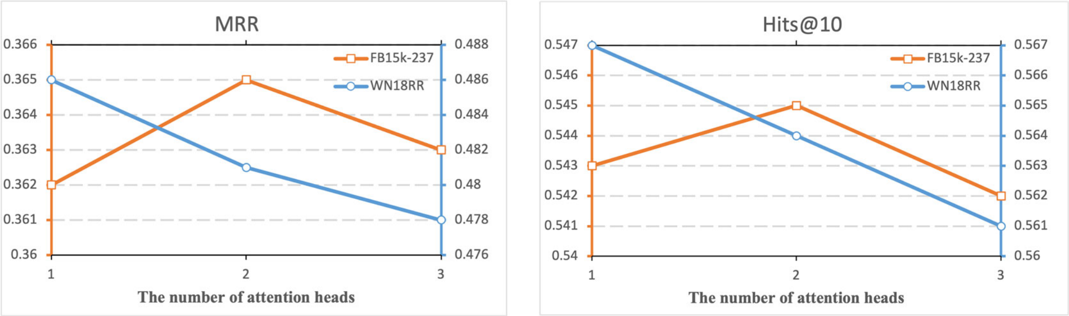

Figure 3. The effect of different numbers of attention heads on experimental results when using multiple heads of attention in a model.

4.6. Ablation Study

As our hierarchical structure outperforms various baseline models with various scoring functions, we investigated the impact of the various modules in the model to provide a comprehensive comparative analysis.

To reflect the different levels of contribution of neighboring relations and entities to the central entity, we designed a relation-level attention mechanism versus an entity-level attention mechanism and added a direct aggregation module for feature information. Then, is it crucial to consider the importance information brought by different entities and relations and use the propagation mechanism of such feature information to deliver messages? To this end, we remove the relation-level attention mechanism, entity-level attention mechanism, and feature information propagation mechanism, respectively, and construct several variants of the model, which are called $\text{HPGAT}_{\text{w/o r−a}}$, $\text{HPGAT}_{\text{w/o e−a}}$, and $\text{HPGAT}_{\text{w/o f−i}}$. For easy comparison, we put the results of the full model with each variant into Table 5 to show them together. Compared with the full model, $\text{HPGAT}_{\text{w/o r−a}}$ obtains the worst results, with a substantial decrease in model performance, while $\text{HPGAT}_{\text{w/o e−a}}$ and $\text{HPGAT}_{\text{w/o f−i}}$ also show different degrees of performance degradation. We analysed the following reasons: (1) The semantic importance between different relations cannot be transmitted and learned autonomously by the neural network due to the missing relation-level attention mechanism. (2) The entity-level attention mechanism is based on relation-level attention, which further divides the past semantic information so that finer-grained neighborhood information can be aggregated. When entities are aggregated with equal feature importance, it results in neighborhood features not being able to participate in the aggregation process in a complete way. (3) The feature information transfer mechanism is based on the assumption that the neural network learning process may pay too much attention to the hierarchical information, and the lack of this process leads to the fact that part of the feature information may be selectively ignored during the learning process, which affects the performance of the model.

Table 5. Result of ablation study.

Model	MRR	Hits@1	Hits@3	Hits@10
w/o r-a	0.355	0.264	0.381	0.531
w/o e-a	0.360	0.271	0.392	0.539
w/o f-i	0.359	0.271	0.391	0.539
HPGAT	0.365	0.276	0.398	0.545

On average, when compared with COMPGCN, our model achieved a significant performance upgrade with all three decoders. Furthermore, our hierarchical structure grants the model access to structural information and potential hierarchical characteristics of entities and relations. Furthermore, the weights assigned to entities and relations accurately convey semantic information in different contexts during the aggregation process, resulting in superior performance compared to the baseline model across various metrics.

5. Conclusions

In this paper, we introduce the Hierarchical Perceptual Graph Attention Network (HPGAT) for the link prediction task. HPGAT utilizes attention mechanisms to capture hierarchical semantic information in complex graphs. Our proposed model utilizes entity-level attention, relation-level attention, and hierarchical aggregation to selectively gather structural information at each level, merge corresponding information features, and weigh them accordingly. HPGAT consolidates entity and relation features, highlighting the feature information of entities at different semantic levels to maximize the exploitation of the graph's structural information. The experiments show the efficacy of our suggested model in predicting links. We plan to explore linkage information with the hierarchical aggregation of entities and further optimized relations in the future.

Author Contributions: Conceptualization, W.H.; methodology, W.H.; software, W.H.; validation, X.L., J.Z., H.L. and W.H.; investigation, W.H.; data curation, W.H.; writing—original draft preparation, W.H.; writing—review and editing, W.H., X.L., H.L. and J.Z.; visualization, W.H.; supervision, W.H.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Projects of Open Cooperation of Henan Academy of Sciences (Grant No. 220901008) and Major Science and Technology Projects of the Ministry of Water Resources (Grant No. SKS-2022029).

Data Availability Statement: The datasets in this paper are available in the following articles: FB15k-237 [30]; WN18RR [16].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.-Y. Collaborative Knowledge Base Embedding for Recommender Systems. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 353–362.
2. Bordes, A.; Chopra, S.; Weston, J. Question Answering with Subgraph Embeddings. *arXiv* **2014**, arXiv:1406.3676.
3. Ma, Y.; Crook, P.A.; Sarikaya, R.; Fosler-Lussier, E. Knowledge Graph Inference for Spoken Dialog Systems. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 5346–5350.
4. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
5. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A Core of Semantic Knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
6. Miller, G.A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *38*, 39–41. https://doi.org/10.1145/219717.219748.
7. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24 August 2014; pp. 601–610.
8. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
9. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28.
10. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Long Papers; Volume 1, pp. 687–696.
11. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
12. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the ICML, Bellevue, WA, USA, 28 June–2 July 2011.

13. Yang, B.; Yih, S.W.T.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
14. Nickel, M.; Rosasco, L.; Poggio, T. Holographic Embeddings of Knowledge Graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30. https://doi.org/10.1609/aaai.v30i1.10314.
15. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 2071–2080.
16. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
17. Jiang, X.; Wang, Q.; Wang, B. Adaptive Convolution for Multi-Relational Learning. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Long and Short Papers; Volume 1, pp. 978–987.
18. Dai Quoc Nguyen, T.D.N.; Nguyen, D.Q.; Phung, D. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. *arXiv* **2018**, arXiv:1712.02121.
19. Socher, R.; Chen, D.; Manning, C.D.; Ng, A.Y. Reasoning with Neural Tensor Networks for Knowledge Base Completion. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Curran Associates Inc.: Red Hook, NY, USA; Volume 1, pp. 926–934.
20. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
21. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903.
22. Vashishth, S.; Sanyal, S.; Nitin, V.; Talukdar, P. Composition-Based Multi-Relational Graph Convolutional Networks. Available online: https://arxiv.org/abs/1911.03082v2 (accessed on 9 January 2024).
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. https://doi.org/10.48550/arXiv.1706.03762.
24. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
25. Cao, J.; Fang, J.; Meng, Z.; Liang, S. Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces *arXiv* **2023**, arXiv:2211.03536.
26. Vashishth, S.; Sanyal, S.; Nitin, V.; Agrawal, N.; Talukdar, P. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton, NY, USA, 7–12 February 2020; Volume 34, pp. 3009–3016.
27. Liao, S.; Liang, S.; Meng, Z.; Zhang, Q. Learning Dynamic Embeddings for Temporal Knowledge Graphs. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, New York, NY, USA, 8 March 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 535–543.
28. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In Proceedings of the the Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Greece, 3–7 June 2018; Gangemi, A., Navigli, R., Vidal, M.-E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 593–607.
29. Nathani, D.; Chauhan, J.; Sharma, C.; Kaul, M. Learning Attention-Based Embeddings for Relation Prediction in Knowledge Graphs. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2018; Korhonen, A., Traum, D., Màrquez, L., Eds.; Association for Computational Linguistics: Florence, Italy, 2019; pp. 4710–4723.
30. Toutanova, K.; Chen, D. Observed versus latent features for knowledge base and text inference. In Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, Beijing, China, 26–31 July 2015; pp. 57–66.
31. Lacroix, T.; Usunier, N.; Obozinski, G. Canonical Tensor Decomposition for Knowledge Base Completion. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 2863–2872.
32. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
33. Balažević, I.; Allen, C.; Hospedales, T. Multi-Relational Poincaré Graph Embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 4463–4473.
34. Chao, L.; He, J.; Wang, T.; Chu, W. PairRE: Knowledge Graph Embeddings via Paired Relation Vectors. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtually, 1–6 August 2021; Long Papers; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Cedarville, OH, USA, 2021; Volume 1, pp. 4360–4369.
35. Zeb, A.; Saif, S.; Chen, J.; Zhang, D. Learning Knowledge Graph Embeddings by Deep Relational Roto-Reflection. *Knowl. -Based Syst.* **2022**, *252*, 109451. https://doi.org/10.1016/j.knosys.2022.109451.
36. Li, Z.; Zhao, Y.; Zhang, Y.; Zhang, Z. Multi-Relational Graph Attention Networks for Knowledge Graph Completion. *Knowl. -Based Syst.* **2022**, *251*, 109262. https://doi.org/10.1016/j.knosys.2022.109262.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.