Article:
Li, C., Peng, X. orcid.org/0000-0001-5787-9982, Zhang, S. et al. (5 more authors) (2020) Modeling relation paths for knowledge base completion via joint adversarial training. Knowledge-Based Systems, 201-202. 105865. ISSN 0950-7051

https://doi.org/10.1016/j.knosys.2020.105865

Modeling Relation Paths for Knowledge Base Completion via Joint Adversarial Training

Chen Li[a,b], Xutan Peng[a,b], Shanghang Zhang[c], Hao Peng[a,b], Philip S. Yu[d], Min He[e], Linfeng Du[a], Lihong Wang*[e]

[a] *Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China*
[b] *State Key Laboratory of Software Development Environment Lab, Beihang University, Beijing, China*
[c] *Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, CP, USA*
[d] *Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA*
[e] *National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China*

Abstract

Knowledge Base Completion (KBC), which aims at determining the missing relations between entity pairs, has received increasing attention in recent years. Most existing KBC methods focus on either embedding the Knowledge Base (KB) into a specific semantic space or leveraging the joint probability of Random Walks (RWs) on multi-hop paths. Only a few unified models take both semantic and path-related features into consideration with adequacy. In this paper, we propose a novel method to explore the intrinsic relationship between the single relation (i.e. 1-hop path) and multi-hop paths between paired entities. We use Hierarchical Attention Networks (HANs) to select important relations in multi-hop paths and encode them into low-dimensional vectors. By treating relations and multi-hop paths as two different input sources, we use a feature extractor, which is shared by two downstream components (i.e. relation classifier and source discriminator), to capture shared/similar information between them. By joint adversarial training, we encourage our model to extract features from the multi-hop paths which are representative for relation completion. We apply the trained model (except for the source discriminator) to several large-scale KBs for relation completion. Experimental results show that our method outperforms existing path information-based approaches. Since each sub-module of our model can be well interpreted, our model can be applied to a large number of relation learning tasks.

Keywords: Joint Adversarial Training, Hierarchical Attention Mechanism, Knowledge Base Completion

1. Introduction

Knowledge Base, which is used to store structured knowledge data, has achieved satisfactory results in many artificial intelligence tasks such as question answering [8], information retrieval [31], and relation extraction [4]. Existing large-scale KBs (e.g., Freebase [3] and WordNet [30]) contain a vast amount of facts about the real world in the form of triples such as (*Head Entity*, `Relation`, *Tail Entity*). However, the growth of edges in KBs is not in proportion to that of entities due to the lack of data collection [5]. In other words, edges in existing KBs are extremely sparse and far from complete. Complementing most of the missing facts manually will cost an unacceptable amount of time and resources. Hence, a large number of Knowledge Base Completion techniques based on analyzing the labeled data have been proposed.

The existing KBC methods can be roughly divided into two types: Knowledge Representation (KR) based methods, and Path Ranking (PR) based methods. KR-based methods, represented by the translation models [6, 37, 25], which embed the knowledge in a KB into a specific continuous vector space and exploit uniform spatial relationships in the projective space to describe the corresponding relations. Based on the representation of the given entity pair, KR-based methods define a score function to rank candidate relations to complete the KB. Meanwhile, there exist Path Ranking (PR) based methods, which enumerate and select valuable paths among the given entity pair as the features to be analyzed. PR-based methods determine the missing relation through training Random Walk (RW) joint probability of selected paths, and also the corresponding relation classifier [22]. PR-based methods and their variants have achieved a satisfactory result on KB completion by optimizing the calculation strategies of path similarity and path selection [23, 27].

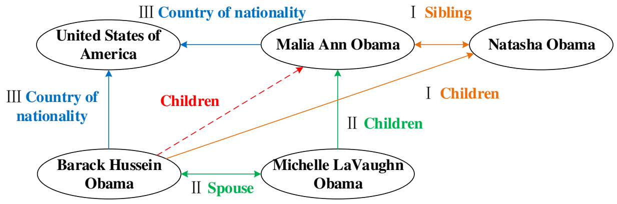


Figure 1: An example of KBC in Freebase. The missing single relation is `Children` (which can be indicated by the red dotted line).

A drawback of the KR-based models is that they only take semantic information implied by the single relations (1-hop paths) into consideration, thus ignoring the interpretation of multi-hop paths among the paired entities [24, 33]. For example, as shown in Figure 1, the missing relation `Children` can be inferred by observing the multi-hop paths between *Barack Hussein Obama* and *Malia Ann Obama* (i.e. paths I (yellow) and II (green)). More generally, the semantic relationship and co-occurrence information is intrinsically built in the single relation and multi-hop paths between the paired entities. Based on this principle, PR-based methods capture the features of the multi-hop paths and calculate the joint probability of each candidate relation. However, PR-based approaches also have their drawbacks, which lie in the existence of noise within multi-hop paths (as illustrated in Figure 1, path III (blue) with two hops is useless for determining the missing relation) as well as the lack of mining the inner relationships between single relation and multi-hop paths [27].

Based on an intuitive assumption that a vast amount of paths between the paired entities generally contain shared/similar semantic and co-occurrence information, this paper presents a unified model that takes both the single relation and multi-hop paths between the given entity pair into full account. Specifically, our model complements the missing single relation by extracting shared information from multi-hop paths. To achieve that, our model leverages Hierarchical Attention Networks (HANs) to encode the inputs and send the features to a downstream feature extraction module, where shared features are automatically captured without any manual selection through Adversarial Training (AT). Among them, the HANs is used to ensure that the features fed to the downstream modules retain the important information in multi-hop paths. The feature extraction module consists of two tightly coupled and parameter-shared neural network structures, one for relation prediction and one for source discrimination. The whole framework is jointly adversarial trained so that the extracted features not only minimize the classification error but also make the source discriminator unable to discriminate their source. Finally, we can get the partial information of a single relation by only taking multi-hop paths as input, so as to determine the missing relation.

The experimental results on FB and WN demonstrate that our method outperforms the state-of-the-art approaches. More generally, since the sub-modules in our model can be well interpreted, our method can transform various tasks (e.g., reasoning and information extraction) into encoding and decoding tasks by defining appropriate adversarial data sources.

The contributions of this paper can be summarized as follows:

- We propose a general approach to mine the intrinsic relationship between single relation and multi-hop paths based on specific downstream tasks.
- We propose a novel KBC approach which outperforms the state-of-the-art methods.
- By visualizing and analyzing the working processes of the components in our model, we interpret their corresponding functions and effects.

The rest of this paper is organized as follows: Section 2 reviews traditional KBC methods and the AT framework. Section 3 details the composition of the whole framework and each of the sub-modules proposed in this paper, and conducts a theoretical analysis. Section 4 introduces the experimental details and analyzes the results of typical downstream tasks. At length, section 5 concludes this paper.

2. Related Work

2.1. Knowledge Base Completion

As mentioned above, the existing methods for KBC mainly include the KR-based models [5, 6, 7, 37, 25] and PR-based models [22, 12, 23, 27].

KR-based models, originating from representation learning, focus on modeling triples in structured [5] or unstructured [7] embeddings. Trans-family models [6, 37, 25] train embeddings based on the assumed spatial positional relationship that $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$. Recent years have seen more elaborate triple modeling methods including: DistMult, which defines a triple scoring method based on matrix operation [14]; ComplEx, which introduces complex space into KR to handle two-way relations [17]. Also, The work of Sun et al. [15] focuses on spatial rotation to optimize triple modeling. In addition, ConvE improves the quality of KR by modeling the relations through convolution methods [13]. Similarly, Shang et al. [41] and Nguyen et al. [42] achieve KBC based on convolutional neural networks. Nguyen et al. [44] introduce capsule neural embedding architecture for learning knowledge embedding to complement the relations. MultiE models triples based on multi-task learning [45]. Moreover, through utilizing canonical tensor decomposition, Lacroix et al. [46] obtain high-quality knowledge embedding for KBC. In general, the KR-based methods vectorize the entities and relations and achieve the purpose of KBC by ranking candidate relations.

In contrast to KR-based methods, PR-based methods generally leverage a RW-based inference technique. Initially proposed by Lao and Cohen [22], PR-based methods determine the missing edges by planning the optimal RW path between entities, and calculating the joint probabilities of the selected paths. Further innovations lay in variant methods which can be summarized as the contextual information/feature extraction of paths [27], the planning of RW [23], and the optimization of path similarity calculation [12].

Recently, some research has begun focusing on incorporating extra information into the completion model. Among them, Lin et al. [24] introduce low-order (2 or 3) path-related information into TransE in three different ways (ADD, MUL, and RNN) and optimize the model. Durán et al. [10] propose an extension of TransE that learns to explicitly model the composition of relations via the addition of their corresponding translation vectors. The work of Xie et al. [39] equips the Trans-family model with a sparse attention, which represents the hidden concepts of relations and transfers statistical strength. Wei et al. [38] combine external memory and the RW strategy to mine the inference formulas. The method proposed by Shen et al. [33] uses a global memory and a controller module to learn multi-hop paths in vector space and infers missing facts jointly without any human-designed procedure. Yoshikawa et al. [43] achieve KBC by combining axiom injection as an inference basis. By constructing rich features, Komninos and Manandhar [47] use a neural network to complete missing relations.

Table 1 lists the time and space complexity of some classical methods. Within the experimental setting of 3 as the upper limit of the paths' hop number, we can conclude that the time

Table 1: Complexity (the number of parameters and the number of multiplication operations in an epoch) of several models. N_e and N_r represent the number of entities and relations, respectively. N_t represents the number of triplets in a knowledge graph. $N_{p,i}$ denotes the number of i-hop paths in dataset. m is the dimension of entity embedding space and n is the dimension of relation embedding space. d denotes the average number of clusters of a relation. l_i is the length of i-hop paths. k is the number of hidden nodes of a neural network and s is the number of slices per tensor. In both datasets of FB15K* and WN18* $N_{p,i} < N_t$ with $N_t/N_{p,i} = 15.7$ and 1.97 respectively.

	Time Complexity	Space Complexity
TransE	$O(N_t)$	$O(N_e m + N_r n)$
TransH	$O(2mN_t)$	$O(N_e m + 2N_r n)$
TransR	$O(2mnN_t)$	$O(N_e m + N_r(m+1)n)$
PTransE(ADD)	$O(N_t)$	$O(N_e m + N_r n)$
PTransE(MUL)	$O(N_t + \sum l_i N_{p,i})$	$O(N_e m + N_r n)$
PTransE(RNN)	$O(N_t + \sum l_i N_{p,i})$	$O(N_e m + N_r n)$
DistMult	$O(2N_t)$	$O(N_e m + N_r n)$
ComplEx	$O(2N_t)$	$O(N_e m + N_r n)$
ConvE	$O(n'N_t)$	$O(N_e m + N_r n')$
RotatE	$O(2N_t)$	$O(N_e m + N_r n)$
Ours	$O(\sum(l_i + 1)N_{p,i})$	$O(N_e m + N_r n)$

complexity of our model on the FB series datasets is less than that of the other methods', and the time complexity on the WN series datasets is close to $O(2N_t)$. With improvements in time complexity, our model's space complexity is similar to that of the others'.

The common drawback of the existing models is the lack of mining the internal semantic relations between single relation and multi-hop paths, which could bring in path-related information for KBC. In this paper, we present a novel method to exploit the intrinsic semantic and co-occurrence relationships between single relation and multi-hop paths, and use them to complete the knowledge base.

2.2. Adversarial Training

Inspired by the two player game, the Generative Adversarial Networks (GANs) consist of two connected sub-modules, in which the discriminator is trained to distinguish if the input sample matches the true data distribution or is generated by the generator while the generator is trained to generate fake samples that maximize the error rate of the discriminator. GANs have been successful in a large number of image-related tasks, and the adversarial framework in GANs has been adapted to fit many other tasks by introducing additional structures. Among them, Han et al. [20] propose a method for learning implicit features by merging the adversarial mechanism into auto-encoder, which realizes the processing of general data. Tadesse and Collons [16] utilize the adversarial framework to unify the semantic information of different sources into word-char embeddings, and have achieved improvements in tasks of sequence labeling. The work of Qin et al. [19] integrates the adversarial mechanism into the screening of training samples, which improves the quality of input samples of distant supervision, and consequently improves the performance of information extraction. Yasunaga et al. [18] analyze the specific effects of the adversarial mechanism in natural language processing in detail and propose a robust model for multi-lingual part-of-speech tagging tasks. In summary, we find that AT can extract shared information between different sources effectively under specific requirements.

3. Methodology

In this section, we introduce the proposed KBC method in detail.

3.1. Problem Definition and Notations

Given a KB containing a collection of triples $\mathcal{T} = \{(h, r, t) \mid h, t \in E, r \in R\}$, where h, t respectively denote the head and tail entity, r denotes the single relation. E and R are the entity and relation sets. A multi-hop path is a sequence of relations $p_i = (r_{i1}, r_{i2}, \ldots, r_{iT_i}), p_i \in \mathcal{P}, r_{ij} \in R, i \in [1, N_p], j \in [1, T_i]$, where p_i is a path in the multi-hop paths set $\mathcal{P}$, and N_p and T_i denote the size of $\mathcal{P}$ and the number of relations in p_i. $\mathcal{P}$ contains all of the paths between (h, t) except the single relation r. The task of KBC is to train a relation classifier based on the labeled data $\{r\} \bigcup \mathcal{P}$ and complement the missing relation r between the given entity pair (h, t).

3.2. Overall Architecture and Components

The overall architecture of the proposed method is shown in Figure 2. The hierarchical attention-based encoders are used to encode the single relation and multi-hop paths as input, and the coupled neural networks are used to extract features. We will describe the detail of different components in the following sections.

3.2.1. Hierarchical Attention-Based Encoders

Given a path $p_i, i \in [1, N_p]$, we first embed the relations into vectors through an embedding matrix $W_t, v_{ij} = W_t r_{ij}$, where r_{ij} denotes the jth hop relation in p_i. Additionally, we add Position Encoding [34] pe_{ij} and direction information d_{ij} after the relation representation to get the final input vector $x_{ij} = [v_{ij}; pe_{ij}; d_{ij}]$. We utilize a Bidirectional Gated Recurrent Unit (Bi-GRU) [1] to get the annotations of the relation sequence, which summarizes the information from both directions.

As illustrated in the left part of Figure 2, we obtain the annotation for each relation r_{ij} in p_i by concatenating the forward hidden state $\overrightarrow{h_{ij}}$ and the backward hidden state $\overleftarrow{h_{ij}}$, i.e. $h_{ij} = [\overrightarrow{h_{ij}}; \overleftarrow{h_{ij}}]$. Among them, $\overrightarrow{h_{ij}}$ (or $\overleftarrow{h_{ij}}$) is computed as

$$\overrightarrow{h_{ij}} = (1 - z_{ij}) \odot h_{i,j-1} + z_{ij} \odot \widetilde{h}_{ij}, \tag{1}$$

where $\overrightarrow{h_{ij}}$ is a linear interpolation between the previous state $h_{i,j-1}$ and the candidate state $\widetilde{h}_{ij}$.

GRU utilizes a gating mechanism to track the states of the sequence. Update gate z_{ij} decides how much past information is kept and how much new information is added, which is computed as

$$z_{ij} = \sigma(W_z x_{ij} + U_z h_{i,j-1} + b_z), \tag{2}$$

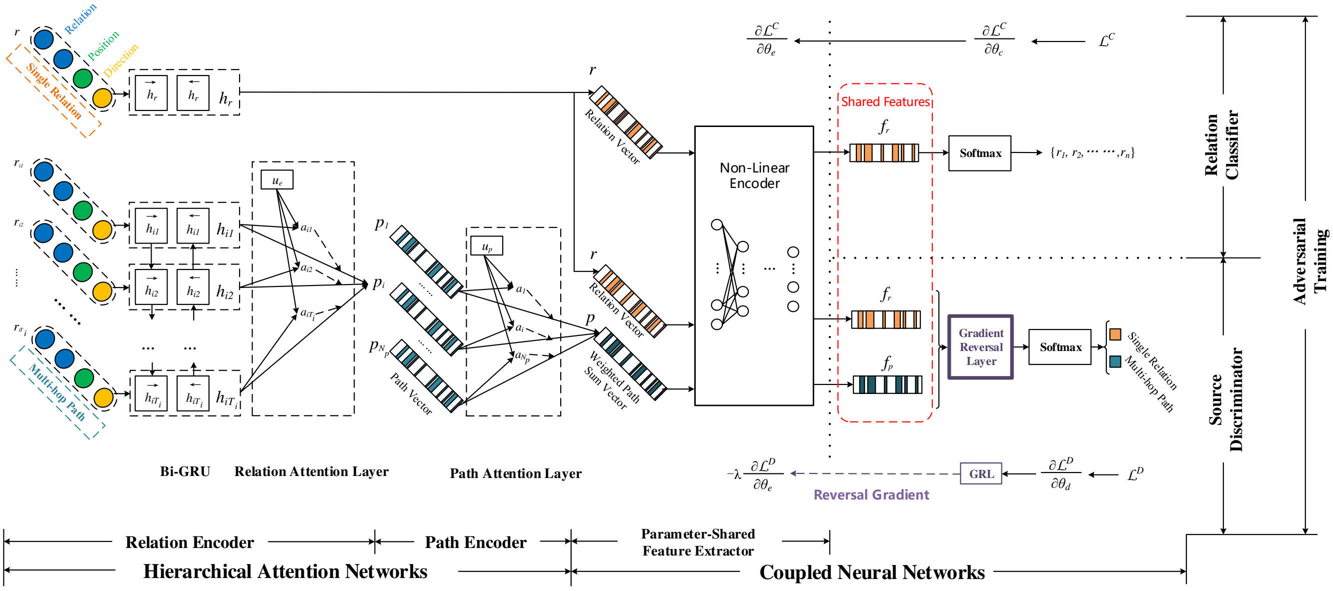


Figure 2: The overview of the architecture of the proposed model.

where $\sigma(\cdot)$ is a sigmod activation function. The candidate state $\widetilde{h}_{ij}$ is computed in a way similar to a traditional RNN:

$$\widetilde{h}_{ij} = \tanh(W_h x_{ij} + reset_{ij} \odot (U_h h_{i,j-1}) + b_h), \tag{3}$$

where $\tanh(\cdot)$ is a tanh activation function.

Reset gate $reset_{ij}$ controls how much the past state contributes to the candidate state $\widetilde{h}_{ij}$. Its update function is as follows:

$$reset_{ij} = \sigma(W_{reset} x_{ij} + U_{reset} h_{i,j-1} + b_{reset}). \tag{4}$$

Different relations may contribute differently to the semantic representation of the multi-hop path. Hence, we introduce the attention mechanism [1] to extract the important relations and aggregate the representation of those meaningful relations to form a specific output. As illustrated in the relation attention layer in Figure 2, we utilize a single attention layer to get u_{ij} as a hidden representation of h_{ij}:

$$u_{ij} = \tanh(W_e h_{ij} + b_e). \tag{5}$$

Specifically, we measure the importance of the relation as the similarity/relatedness of u_{ij} with a context entity vector u_e, and get a normalized importance weight a_{ij} using a softmax function as

$$a_{ij} = \frac{\exp(u_{ij}^\top u_e)}{\sum_{k=1}^{T_i} \exp(u_{ik}^\top u_e)}, \tag{6}$$

where $\exp(\cdot)$ is an exponential function based on the natural constant e. Then, we obtain the representation of the multi-hop path by weighted summing of the relation annotations as follows:

$$p_i = \sum_{j=1}^{T_i} a_{ij} h_{ij}. \tag{7}$$

As shown in the path encoder in Figure 2, to mine the shared semantic features between single relation and multi-hop paths, we use a weighted sum over all the encoded path vectors between entities. As the process and calculation methods are similar to the relation encoder, we omit the detailed description for simplicity. So far, we selectively encode all multi-hop paths in $\mathcal{P}$ into a low-dimensional vector based on HANs.

3.2.2. *Coupled Neural Networks*

In the proposed method, we define two feed-forward architectures to discriminate the source of the extracted feature and its corresponding relation. As shown in the right part of Figure 2, the two neural networks are coupled together by a submodule that extracts features from different input sources. Under the assumption mentioned above, these extracted features are considered as the semantic information shared by the encoded multi-hop paths p and the encoded single relation r. We decompose such coupled neural networks into three parts to detail each sub-module.

i. Parameter-Shared Feature Extractor: As a sub-module shared by two neural networks, the feature f extracted by the feature extractor should minimize the classification error and be source-indiscernible in the meantime. In the proposed model, the feature mapping is accomplished by several feed-forward layers, which are similar to the encoder in stacked AutoEncoder [36]. Because this sub-module is shared by the follow-up discriminator and classifier, all of the parameters θ_e in the feature extractor $\mathcal{E}(x; \theta_e)$ are updated based on the gradients from downstream parts (as illustrated in the upper $\mathcal{L}^C$ and bottom $\mathcal{L}^D$ in Figure 2). To improve the ability of the feature extractor to resist noise and remain sparse, we add sparse constraints in the loss function. After which, the loss function of the feature extractor is defined as

$$\mathcal{L}^E = \mathcal{L} + \beta \sum_j KL(\rho \| \hat{\rho}_j), \tag{8}$$

where $\mathcal{L}$ is the downstream loss (i.e. the combination of $\mathcal{L}^C$ and $\mathcal{L}^D$), β denotes the constraint rate, $KL(\cdot)$ is the KL divergence, and ρ and ρ_j represent the expected activation of neurons and average activation degree on the training set.

In general, for the source discriminator, the feature extractor can be considered as a query of 'what are the source-shared features?' Thus from the classifier's perspective, it can be regarded as a query of 'what are the valuable features for determining relation?' Furthermore, it also stands for a high-level representation of a fixed query in the whole model of 'what are the pivots between a source discriminator and relation classifier?' More theoretically, to ensure that the features extracted by the feature extractor can be obtained from the single relation and the multi-hop paths, and can effectively correspond to the true single relation, this paper selects a coupled adversarial structure to achieve the above two constraints. Among them, the source discriminator confuses features from different sources by providing a reversal gradient, so that the features obtained by the feature extractor from different sources obey a similar distribution. The relation classifier can still accurately correspond to the unique correct relation by the similar features obtained from the single relation and the multi-hop path through the classification error constraint.

ii. Source Discriminator: The source discriminator is a sub-module exploited to discriminate the source of input features. Its discriminant results can be used to optimize the performance of the feature extractor. As illustrated in the bottom right part of Figure 2, we treat the shared features $f \in \{f_r, f_p\}$ as a credential that determines the source of the input features which is computed as

$$z_i = W_{di} f + b_{di}, y_i = \frac{\exp(z_i)}{\sum_{j=1}^{2} \exp(z_j)}, \tag{9}$$

where $z_i, i \in [0, 1]$ is the independent probability of each data source, y_i is the normalized probability for different data sources. The goal of the source discriminator is to minimize the cross-entropy loss within all data distribution:

$$\mathcal{L}^D = -\frac{1}{2N_s} \sum_{k=1}^{2N_s} (\hat{y_k} \ln y_k + (1 - \hat{y_k}) \ln(1 - y_k)), \tag{10}$$

where $\hat{y_k} \in \{0, 1\}$ denotes the real source of sample k and N_s is the number of positive samples.

To ensure that the features extracted by the feature extractor are source-indiscernible, we introduce *Gradient Reversal Layer* (GRL) [9] between the feature extractor and the source discriminator, as shown in the bottom right part of Figure 2. The GRL achieves gradient reversal through multiplying the gradient by a certain negative constant during the back propagation-based training, which guarantees similar feature distributions over the two sources. Specifically, GRL acts as an identity transform during the forward propagation, but takes the gradient from subsequent level, multiplies it by $-\lambda$, and passes it to the preceding layer. In our framework, we treat the GRL as a "pseudo-function" $R_\lambda(x)$ defined by the following equations, describing its forward- and back-propagation behaviors:

$$R_\lambda(x) = x, \frac{\partial R_\lambda(x)}{\partial x} = -\lambda I, \tag{11}$$

where I is an identity matrix and λ is the adaptation rate. Hence, the independent possibility z_i has to be rewritten as

$$z_i = W_{di} R_\lambda(f) + b_{di}. \tag{12}$$

Mathematically, we treat the source discriminator as $\mathcal{D}(R_\lambda(f); \theta_d)$ and the feature extractor as $\mathcal{E}(x; \theta_e)$, where θ_d and θ_e are corresponding parameters. As illustrated in Figure 2, the θ_d is optimized to enhance the capability of the source discriminator to distinguish different data sources, while θ_e is trained to fool the discriminator based on the reversal gradient $-\lambda \frac{\partial \mathcal{L}^D}{\partial \theta_e}$.

iii. Relation Classifier: This sub-module is used to determine the missing relation between the given entity pairs. It outputs the probability of each candidate relation in relation set R. As illustrated in the top right part of Figure 2, we treat the output from the feature extractor as the representative features to determine the single relation and feed it to the softmax layer for relation classification. Because the calculation method of the softmax layer is similar to that of the corresponding part in the source discriminator, we omit it. The loss function for the relation classifier can be written as

$$\mathcal{L}^C = -\frac{1}{N_s} \sum_{k=1}^{N_s} \sum_{i=1}^{N_c} (\hat{y_i} \ln y_{k\_i}), \tag{13}$$

where $\hat{y_i} \in \{0, 1\}$ and y_{k_i} are the ground truth and output result respectively.

3.2.3. Regularization

In order to avoid over-fitting, the squared Frobenius norm and squared ℓ_2 norm for different weight W_d, W_c and bias b_d, b_c are added as

$$\mathcal{L}^R = \|W_d\|_F^2 + \|b_d\|_2^2 + \|W_p\|_F^2 + \|b_p\|_2^2, \tag{14}$$

where $\|\cdot\|_2$ and $\|\cdot\|_F$ respectively denote the ℓ_2 norm of a vector and Frobenius norm of a matrix.

3.3. Joint Adversarial Training

The overall objective function of the proposed model is a combination of the losses of its components.

$$\mathcal{L}^{Total} = \mathcal{L}^D + \mathcal{L}^C + \beta \sum_j KL(\rho \| \hat{\rho}_j) + \rho_r \mathcal{L}^R, \tag{15}$$

where ρ_r is the regularization parameter. To minimize $\mathcal{L}^{Total}$, we use a joint AT strategy to iteratively update all sub-modules.

Before AT, we pre-train the components with specific methods to avoid possible problems (e.g., gradient vanishing, and gradient instability) [35]. The implementation details (including initialization, determining parameters, and pre-training) of each component will be detailed in the follow-up section.

During AT, the source discriminator[1] and the relation classifier can be directly trained to minimize their loss by using

[1] To ensure the performance of the source discriminator, we manually adjust the input distribution in each mini-batch to balance the number of inputs from the two sources.

stochastic gradient descent (SGD) based method as

$$\theta_c := \theta_c + \mu\frac{\partial \mathcal{L}^C}{\partial \theta_c}, \theta_d := \theta_d + \mu\frac{\partial \mathcal{L}^D}{\partial \theta_d}, \tag{16}$$

where θ_c and θ_d denote the parameters in the classifier and discriminator respectively, and μ is the learning rate. As the upstream module, the feature extractor has multiple gradient sources. Hence, as illustrated in the bottom part of Figure 2, reversed gradient (through passing the GRL) from the discriminator and back-propagation gradient from the classifier are used to update the parameters by fine-tuning as

$$\theta_e := \theta_e + \mu\frac{\partial \mathcal{L}^E}{\partial \theta_e}. \tag{17}$$

In particular, in order to ensure the performance of AT, we balance the distribution of gradient source in each batch.

3.4. Analysis of Model Perspectives

From the perspective of the whole model, the motivation of our model is to ensure that the shared features extracted from different sources are valuable for completing missing relations. In the following, we propose some alternative perspectives to facilitate understanding of our model.

Information Theory: Consider $X_r \sim q(X_r)$ is the distribution of r, q unknown, $X_{f_r} = f_{\theta_e}(X_r)$. It can easily be shown that minimizing the expected classification error and discrimination error amounts to respectively maximize a lower bound on mutual information $I(X_{f_r}; X_r)$ and $I(X_{f_r}; X_{f_p})$. Because downstream tasks are independent of each other, our model can thus be justified by the objective that $f_{\theta_e}(\cdot)$ captures as much source-indiscernible information as possible for relation classification, i.e. maximizing a lower bound on $I(X_r; X_{f_p})$.

Manifold Learning: We treat f as a low-dimensional manifold, and $f_{\theta_e}(\cdot)$ is a stochastic operator learnt to map r, p to f_r, f_p which tends to go from lower probability points to high probability points, generally on or near the manifold. Our model can thus be seen as a way to define and learn a manifold, and the intermediate representation $Y = f_{\theta_e}(X)$ can be interpreted as a coordinate system for points on the manifold. More generally, one can think of $Y = f_{\theta_e}(X)$ as a representation of X which is well suited to capture the **valuable** (not **main**) variations in the input, i.e. on the manifold. When additional criteria (i.e. source indiscrimination) are introduced in the learning model, one can no longer directly view $Y = f_{\theta_e}(X)$ as an explicit low-dimensional coordinate system for points on the manifold, but it retains the property of capturing the specific factors of source-shared variation in the data.

Generative Model: As our model takes an input vector $x \in \{r\} \bigcup \{p\}, x \in \mathbb{R}^d$, and maps it to features $f \in \{f_r\} \bigcup \{f_p\}, f \in \mathbb{R}^{d'}$ through feature extractor $f = f_{\theta_e}(x)$, parameterized by θ_e. Depending on the different downstream tasks, feature f is then mapped back to the 'reconstructed' outputs, i.e. relation r (through relation classifier) and source (through source discriminator) by $y_c = g_{\theta_c}(f_r)$ and $y_d = h_{\theta_d}(f)$. Each training $x^{(i)}$ is mapped to a corresponding $f^{(i)}$, an output relation y_c (only for x_r) and a source y_d. Since the downstream tasks are pre-trained, the parameters of the model are optimized to minimize the *average reconstruction error* and maximize the *source discrimination error*:

$$\theta_e^\star = \arg\min_{\theta_e}(\frac{1}{m}\sum_{i=1}^{m} L(x^{(i)}, y_c^{(i)}) - \frac{1}{n}\sum_{j=1}^{n} L(x^{(j)}, y_d^{(j)})), \tag{18}$$

where L is a loss function, $m = |\{r\}|, n = |\{r\} \bigcup \{p\}|$.

As mentioned in Section 3.2.2, L is the *cross-entropy*. i.e. $L = L_{\mathbb{H}}$, then:

$$\theta_e^\star = \arg\min_{\theta_e} \mathbb{E}_{q^0(X)}(L_{\mathbb{H}}(X, Y_c) - L_{\mathbb{H}}(X, Y_d)), \tag{19}$$

where $q^0(X)$ denotes the empirical distribution associated to training inputs, $\mathbb{E}_{p(X)}[f(X)] = \int p(x)f(x)\mathrm{d}x$ is the expectation of $f(X)$.

Specifically, we recover the training criterion for our model from a generative model perspective. Then the goal of our model is to learn a constrained joint probability distribution, i.e. the co-occurrence probability of the input data X, the source-indiscernible feature f, and the relation r. Consider the generative model $p(X, f, Y_c, \bar{Y}_d) = p(Y_c, \bar{Y}_d)p(X|Y_c, \bar{Y}_d)p(f|X))$, where $\bar{Y}_d$ denotes the wrong source label. We can show that training the feature extractor as described above is equivalent to maximizing a variational bound on that generative model:

$$\begin{aligned} &\arg\max_{\theta_e} \mathbb{E}_{q^0(X)}[\log p(X, f, Y_c, \bar{Y}_d)] \\ &= \arg\max_{\theta_e} \mathbb{E}_{q^0(X)}[\log p(Y_c, \bar{Y}_d)p(X|Y_c, \bar{Y}_d)p(f|X))]. \end{aligned} \tag{20}$$

$p(Y_c, \bar{Y}_d)$ is a uniform prior probability and $q^0(X|f) \propto q^0(f|X)q^0(X)$, then

$$\begin{aligned} &\arg\max_{\theta_e} \mathbb{E}_{q^0(X)}[\log p(Y_c, \bar{Y}_d)p(X|Y_c, \bar{Y}_d)p(f|X))] \\ &= \arg\max_{\theta_e} \mathbb{E}_{q^0(X)}[\log p(X|Y_c, \bar{Y}_d)] \\ &= \arg\max_{\theta_e} \mathbb{E}_{q^0(X)}[\log(p(X|Y_c = g_{\theta_c}(f_r)) \cdot p(X|\bar{Y}_d = h_{\theta_d}(f)))] \\ &= \arg\min_{\theta_e} \mathbb{E}_{q^0(X)}(L_{\mathbb{H}}(X, Y_c) - L_{\mathbb{H}}(X, Y_d)). \end{aligned} \tag{21}$$

Obviously, this is consistent with our training goals, i.e. Equation 19.

4. Experiments

We evaluate the proposed method empirically in terms of classification effect and model interpretability.

4.1. Datasets

Since our method requires an abundant amount of multi-hop paths between entities, we use standardized FB15K and WN18 as experimental datasets [6]. Among them, FB15K is a highly dense sub-graph captured from Freebase, which contains triples composed by two entities and a relation. WN18 is a set of linguistic triples obtained from WordNet, which is a lexical English dictionary containing linguistic relation between words, e.g., `hypernym`, `hyponym` and `meronym`. We also set up a delicate sub-graph with some high-quality, intuitively inferable

Table 2: Statistics of the five datasets. Since we only retain entity pairs that contain multi-hop paths, there are fewer entities than in the original datasets.

	FB15K*	FB15K-237*	FBe30K*	WN18*	WN18RR*
#Entities	13126	10463	19835	21569	21562
#Entity Pairs	37785	18473	64732	77396	77092
#Relations	1345	237	1493	18	11
Average Multi-hop Paths (*length* $\leq$ 3) between Each Entity Pair	11.62	9.34	12.85	8.63	8.59

sub-graphs[2] from FreebaseEasy [2] as the third experimental dataset (namely FBe30K), to further validate the hierarchical attention mechanism in a relatively controllable context. Furthermore, we also use FB15K-237 and WN18RR as experimental data, which is a filtered version of FB15K and WN18 [29, 13] where reciprocal triples are removed. To train the model, we select the entity pairs with single relation in between and having connecting multi-hops paths, and divide them into training, validation, and test sets by 8:1:1. Note that we add annotation * to emphasize that our datasets have been restructured. Table 2 details the statistical information of the datasets.

4.2. Experimental Setup

We evaluate the the performance of the models by two widely-used link prediction testing protocols: the Mean Rank (MR) of the original triple among corrupted ones, and the proportions of valid entities ranked in top 1%, 3%, and 10% (Hit@1, Hit@3, Hit@10). Following [6], we also report *filtered* results to avoid underestimations caused by corrupted triples that are in fact valid.

4.2.1. Baseline Methods

We select three groups of models and compare them in the reconstructed datasets[3].

Triple information based methods (Group **T**)[4]: TransE models relations between entities by interpreting them as translations operating on the low-dimensional embeddings of the entities [6]. TransH models a relation as a hyperplane together with a translation operation on it and proposes an efficient method to construct negative examples to reduce false negative labels in training [37]. TransR builds entity and relation embeddings in separate entity and relation spaces and learns embeddings by first projecting entities from entity space to corresponding relation space, and then building translations between projected entities [25]. CTransR is cluster-based TransR, which clusters diverse head-tail entity pairs into groups and learns distinct relation vectors for each group [25].

DistMult implements triple modeling by defining triple as a matrix operation [14]. ComplEx introduces complex space to optimize single relationship modeling [17]. ConvE uses 2D convolution to model the relationship between entity pairs [13]. RotatE learns knowledge embedding through relational rotation in complex space [15].

Triple and Path information based methods (Group **T&P**): rTransE learns to explicitly model the composition of relationships via the addition of their corresponding translation vectors [10]. PTransE considers relation paths as translations between entities for representation learning and addresses problems of path reliability and semantic composition [24]. Different methods in multi-hop paths modeling can be divided into (RNN 2-step), (ADD 2-step), and (MUL 2-step).

Path information based methods (Group **P**): B-PR considers binarized path features for learning PR-classifier [11]. SFE-PR is a PR-based method that uses breadth-first search with Subgraph Feature Extraction (SFE) to extract path features [11]. In particular, SFE-PR uses only PR-style features for feature matrix construction. Ours is the model proposed in this paper.

4.2.2. Implementation Details

As we found that various dimensions of relation embeddings and PE resulted in similar trends, only experimental results for 100 dimension relation embeddings and 5 dimension PE will be reported here. To construct path set $\mathcal{P}$, we utilize the shortest path principle $min\{3, n\}$ and RW strategy [23] respectively. We initialize the input embeddings with two different pre-trained embeddings (i.e. Word2Vec [28] and TransE [6]), and randomly initialize other model parameters, e.g., attention vectors, feature extractor and softmax layers of relation classifiers and source discriminators.

Each module is pre-trained after initialization, in which the training target of the whole model (except source discriminator) is to minimize the classification error of the relation classifier. The source discriminator is trained from scratch after the classifier converges. The hyperparameters of the model are chosen by maximizing Mean Rank on the validation set: the regularization weight ρ_r is set to 0.05, the constrain rate β is set to 0.01 and the expected activation degree ρ is set to 0.05. The adaptation factor λ in GRL is gradually changed from 0 to 1 during the training process as $\lambda = \frac{2}{1+\exp(\frac{\gamma \cdot n_c}{nN_s})} - 1$ [9]. Our model is optimized with the SGD over shuffled mini-batches with momentum rate 0.95, learning rate is decayed as $\eta = \frac{0.005}{(1+\frac{\gamma \cdot n_c}{nN_s})^{0.5}}$, where n_c and n respectively denote the number of trained samples and iterations, and γ is set to 10. The batch sizes for the source discriminator and relation classification are both 100, and the batch for source discriminator coming from different sources equally.

[2] The sub-graphs mainly include triples whose relations can be intuitively infer ed, e.g., `Children`, `Spouse`.

[3] We obtain the experimental results of all baseline methods by rerunning their codes.

[4] We only use the selected entity pair (i.e. triple) for training, so there is a gap between our results and those in the original papers.

Table 3: Comparison of different methods for relation classification in FB15K*, FB15K-237*, and FBe30K*. Higher Hits@n and lower Mean Rank indicate better predictive performance.

	Model	FB15K*				FB15K-237*				FBe30K*			
		MR(*filter*)	Hits@1	Hits@3	Hits@10	MR(*filter*)	Hits@1	Hits@3	Hits@10	MR(*filter*)	Hits@1	Hits@3	Hits@10
T	TransE	2.71	89.14	90.70	91.27	3.54	83.88	85.96	87.17	2.67	89.63	90.83	91.51
	TransH	2.50	90.06	91.63	92.43	3.34	84.64	87.08	88.21	2.48	90.37	91.87	92.52
	TransR	2.33	91.25	92.75	93.27	3.19	85.42	87.71	89.08	2.32	91.58	92.67	93.34
	CTransR	2.10	92.30	93.86	94.54	3.11	86.09	88.18	89.43	2.09	92.56	93.92	94.51
	DistMult	1.61	94.71	96.43	97.19	2.63	88.47	90.90	91.98	1.57	94.91	96.72	97.31
	ComplEx	1.59	94.82	96.63	97.23	2.57	88.49	90.97	92.34	1.56	95.73	96.75	97.33
	ConvE	1.46	95.63	97.34	97.84	2.40	89.40	91.84	93.26	1.45	95.84	97.30	97.98
	RotatE	1.44	95.67	97.75	97.98	2.36	89.77	92.01	93.25	1.36	**96.83**	97.64	98.42
T&P	rTransE	1.87	93.51	95.17	95.71	2.77	87.69	90.10	91.26	1.86	94.06	95.21	95.73
	PTransE(MUL 2-step)	1.93	93.04	94.88	95.42	2.95	86.98	88.99	90.30	1.92	93.38	94.86	95.46
	PTransE(RNN 2-step)	1.58	94.74	96.64	97.29	2.52	88.98	91.24	92.62	1.57	95.21	96.71	97.28
	PTransE(ADD 2-step)	1.50	95.27	97.08	97.65	2.43	89.56	91.78	93.05	1.50	95.85	97.11	97.68
P	B-PR	2.52	89.72	91.62	92.35	3.46	84.36	86.48	87.54	2.51	90.49	91.73	92.38
	SFE-PR	2.34	90.87	92.69	93.24	3.25	85.05	87.49	88.75	2.32	91.32	92.67	93.36
	Ours (Word2Vec&Shortest)	2.13	91.82	93.75	94.40	2.86	87.37	89.67	90.73	2.09	92.52	93.93	94.52
	Ours (Word2Vec&RW)	1.98	92.78	94.45	95.14	2.74	87.78	90.26	91.38	1.94	93.38	94.74	95.35
	Ours (TransE&Shortest)	1.50	95.10	97.13	97.68	2.49	88.98	91.32	92.80	1.43	96.05	97.39	98.07
	Ours (TransE&RW)	**1.37**	**95.81**	**97.78**	**98.37**	**2.32**	**89.89**	**92.28**	**93.65**	**1.33**	96.55	**97.97**	**98.50**

4.3. Experimental Result and Analysis

We compare our method with baselines in two aspects as discussed below.

4.3.1. Classification Effect

Table 3 and Table 4 show the comparative results of classification effect on FB and WN respectively. We can observe that our method outperforms the other two groups of baseline methods significantly, except that Hits@1 is slightly lower than RotatE. With the use of different pre-trained relation embeddings and a more reasonable path selection algorithm, the performance of our model is enhanced greatly.

In contrast to the methods in Group **T**, we find that our method (TransE&RW) improves the classification effect in both MR and Hits@ 3, 10. Furthermore, despite the fact that rTransE and PTransE in Group **T&P** introduce path information into original TransE by using additional translation vectors, the overall effect is still not as good as our model's. This indicates that although the introduction of path information can significantly improve the classification effect, its stage and strategy can affect the final results.

By observing the performances on different datasets, we can find that the prediction effects of all models (especially B-PR and SFE-PR) become worse after removing reverse relations. However, since our model can handle multi-hop paths efficiently, the loss of a one-hop reverse path has less impact on the final results.

To investigate the performance of our model in detail, we analyze the performance of our model with different relation frequency, as shown in Figure 3. Since the overall distribution of relations in the five datasets clearly shows a long tail phenomenon (the statistics of single relations on datasets are shown in Figure 4), we divide the relations into three buckets [39]. With each bucket, we compare our method (TransE&RW) with TransE and PTransE (ADD 2-step). The results show that introducing path information can improve the classification effect in all buckets, and our method is significantly better than TransE and PTransE(ADD 2-step) in dealing with low-frequency relations.

Table 4: Comparison of different methods for relation classification in WN18* and WN18RR*. Because most of Hits@3 and Hits@10 for WN datasets are exceed 98%, we only report Hits@1.

	Model	WN18*		WN18RR*	
		MR(*filter*)	Hits@1	MR(*filter*)	Hits@1
T	TransE	1.99	92.88	1.97	92.86
	TransH	1.73	94.04	1.70	94.40
	TransR	1.56	94.89	1.57	95.04
	CTransR	1.54	95.17	1.54	95.22
	DistMult	1.25	96.60	1.24	96.82
	ComplEx	1.26	96.64	1.22	96.69
	ConvE	1.17	97.23	1.13	97.17
	RotatE	1.16	97.31	1.10	97.31
T&P	rTransE	1.41	95.72	1.40	95.77
	PTransE(MUL 2-step)	1.56	94.75	1.53	94.96
	PTransE(RNN 2-step)	1.38	95.82	1.36	96.07
	PTransE(ADD 2-step)	1.20	96.97	1.19	96.88
P	B-PR	1.85	93.61	1.87	93.49
	SFE-PR	1.79	93.93	1.82	93.53
	Ours (Word2Vec&Shortest)	1.49	95.42	1.47	95.60
	Ours (Word2Vec&RW)	1.41	95.72	1.42	95.95
	Ours (TransE&Shortest)	1.18	97.15	1.15	97.15
	Ours (TransE&RW)	**1.13**	**97.39**	**1.07**	**97.44**

Combining the analysis performed over KBs and the statistic information in Table 2, we can conclude that the denser (the ratio of entities to relations) and the larger (the amount of entities and relations) the KB becomes, the better our method outperforms baselines. This means that our method is better suited for relation completion of large-scale and dense KBs.

4.3.2. Parameter Study

In order to measure the impact of fluctuations in the hyperparameters of our model on KBC performance, we perform the following experimental analysis for different hyperparameters.

Path Length: We set the maximum path length (i.e. maximum hop limit) of the selected multi-hop paths between the paired entities in the range of 2 to 4 to observe the impact of path length on model performance. The computational complexity dramatically raises with the increase of maximum path length allowed, so we stop at 4 which still takes acceptable

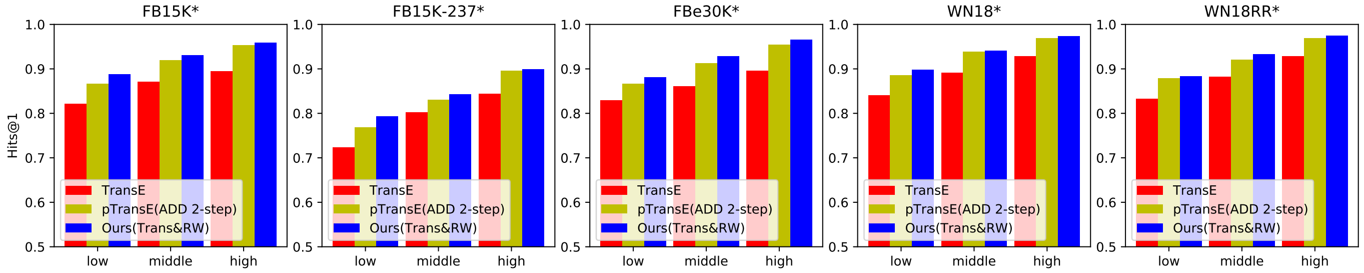
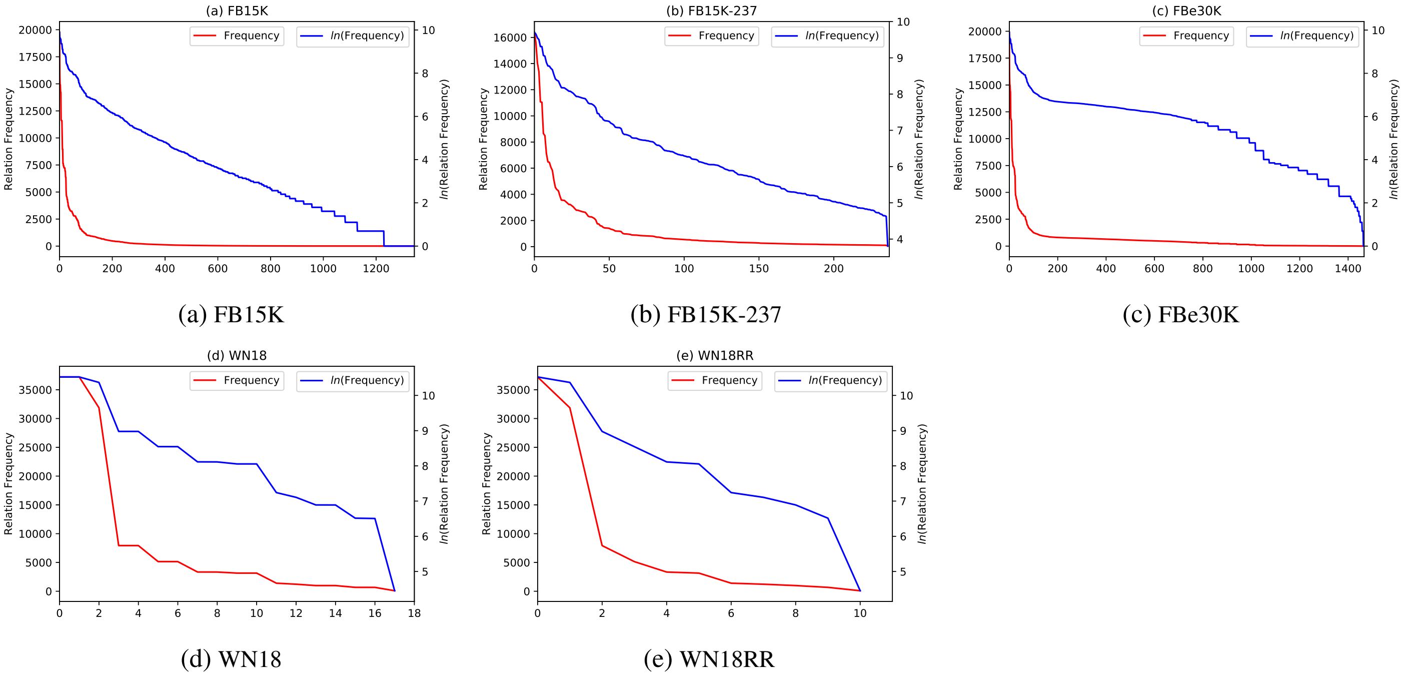


Figure 3: The relation completion results of the three methods in different frequency relations. We chose PTransE (which also introduced path information) and TransE (which is our initial inputs) to compare our method. (low, middle, high) means the distribution of relation frequency.



(a) FB15K

(b) FB15K-237

(c) FBe30K

(d) WN18

(e) WN18RR

Figure 4: Frequencies and log frequencies of single relations in three datasets. The x-axis is single relations sorted by frequency.

computational time on our reduced datasets. The experimental results shown in Figure 5 demonstrate that setting the parameter to 3 achieves significantly better results that setting it to 2. However, as we allow longer path to be sampled (e.g. length of 4), prediction accuracy is not further improved, instead it is weakened. We believe that this is because as the maximum path length increases, the number of sampled multi-hop paths increases rapidly, among which the proportion of noisy/useless paths also grows greatly. Through observations we have noticed that a 5-hop path can almost connect any two entities in our datasets, which definitely introduces a large amount of noise. We also noticed that there are only 18 kinds of relations in the WN, which results in a higher sensitivity to changes in path length limitations, as shown in Figure 5.

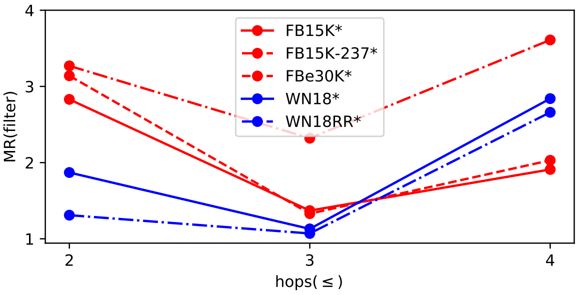


Figure 5: Comparison of different hop upper limit selection. The lower the MR(filter), the better.

Embedding Dimension: As a hyperparameter of KB embedding, the dimension of the embeddings may influence the semantic information contained in the input relations and paths, which may then bring variance to our model. To verify this, we set the embedding dimension to 50, 100, 200, 500, and 1000 respectively and test our model on FB15k-237 and WN18RR[5]. The experimental results in Figure 6 illustrate that the model

[5] The varying dimension does not include the position embedding and direction embedding.

Table 5: Comparison of different feature extraction methods in FB15K.

Method	Trainable	Linear	MR	Hits@1	Hits@3	Hits@10
Non-Linear Encoder (multi-layers)	Yes	No	**1.37**	**95.81**	**97.78**	**98.37**
Non-Linear Encoder (single layer)	Yes	No	2.13	92.65	94.24	95.64
Principal Component Analysis	No	Yes	4.68	78.47	80.33	80.95
Linear Discriminant Analysis	No	Yes	4.57	79.06	80.89	81.54
Locally Linear Embedding	No	No	4.05	81.92	83.72	84.26

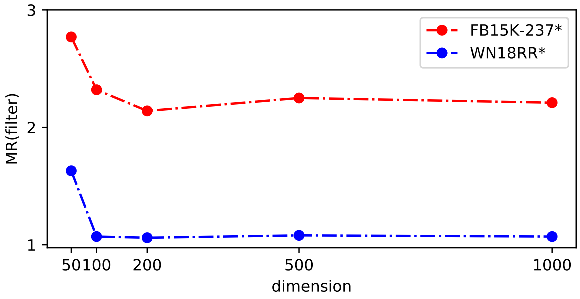
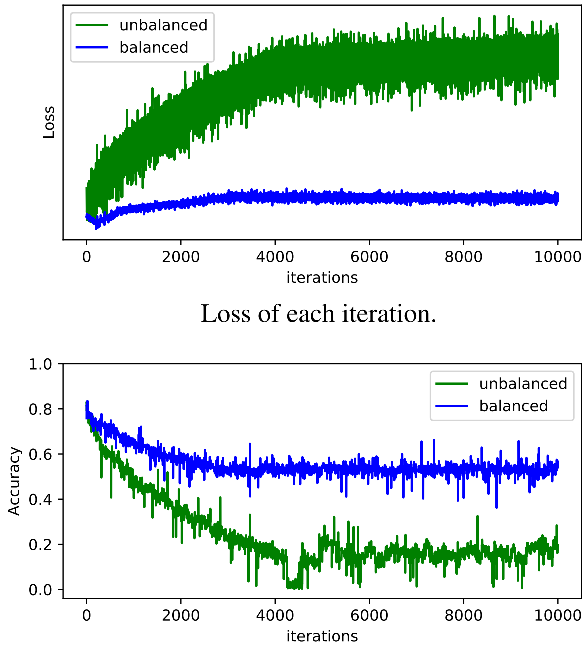


Figure 6: Comparison of different knowledge embedding dimension. The lower the MR(filter), the better.



Loss of each iteration.

Classification accuracy.

Figure 7: The loss and classification accuracy (with GRL) training on the FB15k* varies with iteration.

is sensitive to the variance of embedding dimension when it is low, and becomes insensitive to the increasing dimension as it increases. In particular, the inflection point of the WN's curve takes place earlier than that of the FB'S, which we believe is because fewer relations in FB lets it require fewer dimensions to distinguish different semantic information between the relations.

Data Balance: The source discriminator in our model plays an important role in ensuring that the features extracted by the feature extractor are the shared ones between the single relations and multi-hop paths. However, the training process is hindered by uneven distribution of the two sources as the number of multi-hop paths greatly exceeds that of the single relations. To overcome this, we manually adjust the distribution of samples from the two sources in each mini-batch (of size 236) to balance the proportion. The loss of the source discriminator through iterations before and after the data balance adjustments are shown in Figure 7. The results show that the source discriminator (fed with balanced data) can eventually meet the expected classification accuracy of approximately 0.5, indicating that the source can not be distinguished. Notice that data balance also exists in the relation classifier to tackle similar situations in learning the low-frequency relations.

4.3.3. Model Interpretability

To validate that our model can select the informative multi-hop paths and crucial relations in them, we choose sample triples and visualize the intermediate results of the hierarchical attention layers in Table 6, Table 7, and Table 8 as a case study. The single relations in the tables that should be complemented are: `/location/location/people_born_here`, `/location/country/languages_spoken`, and `/people/person/education./education/education/institution`. By observing the weights of different paths and the relations in one path, we can find that the crucial paths and relations are given higher attention weights. For example, as illustrated in Figure 6, relations `place_of_birth`$^{-1}$, `place_lived/location`$^{-1}$, and `nationality`$^{-1}$ descent in importance for determining the missing relation `people_born_here`. Correspondingly, paths containing them are given decreasing weights. All three relations are given relatively higher weights in their paths, yet their weights decline with decreasing importance. The above results indicate that the HANs in our model can effectively determine the importance of paths and encode them into low dimensional vector by organically combining the semantic information contained in the composing relations.

To further demonstrate this, we randomly select three different single relations and subsequently the most important multi-hop paths that decide them in FB15K*. We use PCA to reduce the dimensions of their embeddings and visualize the result in Figure 8, which shows that the weighted sum of the multi=hop paths is closer to the missing single relation than each multi-

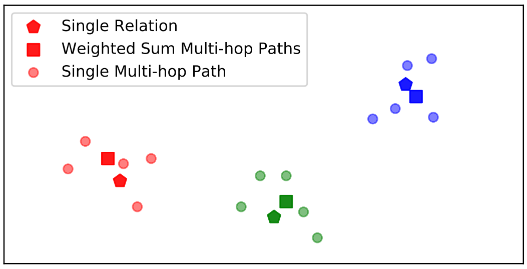
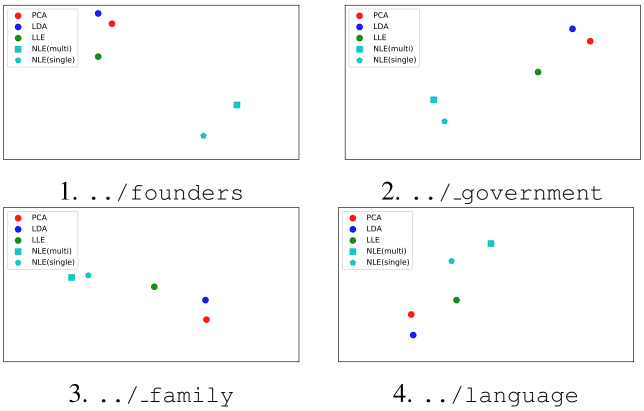


Figure 8: The visualization of dimension reduction results of the single relation r in FB15K*, single path codings p_i (top 5) and weighted sum codings p. Three colors represent three different relations respectively.



1. ../founders

2. ../_government

3. ../_family

4. ../language

Figure 9: The visualization of examples in FB15K* for dimension reduction using different algorithms. We find that in the vast majority of dimension reduction results, the outputs of our feature extractor clearly deviate from other methods (which retain original information as much as possible).

hop path alone, indicating a semantically closer relationship between them. It effectively verifies that the HANs can effectively incorporate the semantic information of the multi-hop paths.

We compare our feature extractor with several common dimension reduction algorithms on FB15K* [26, 32]. Experimental results illustrated in Table 5 show that our method has a higher classification performance. To further analyze the difference between our method and dimension reduction algorithms, we visualize the features extracted with different methods in a common vector space as shown in Figure 9. We find that our feature extractor, which is driven by the downstream submodules of the proposed model, gets completely different results from the traditional dimension reduction methods (i.e. PCA, LDA, and LLE), which aim at retaining information of the original data as much as possible, while the results of the latter do not show much variance. This indicates that the upper-stream HANs has reduced and discarded some of the noisy/useless information for downstream tasks, which coincides with our description in Section 3.4. Fundamentally, the visualization results validate our assumption to some extent, i.e. there exists shared semantic information and noisy/useless information between the single relation and multi-hop paths.

5. Conclusion

In this paper, we propose a novel method for encoding and extracting the shared features between the single relation and multi-hop paths among paired entities based on the simple hypothesis that the connecting relation and paths generally share a vast amount of semantic information. We utilize it for KBC without any manual pivots. Experiments on five knowledge base datasets demonstrate the effectiveness of our model. Furthermore, by independently verifying each sub-module, we show that through AT, the HANs and coupled neural networks in our model has successfully captured the indistinguishable yet valuable features and used it to complete the missing relations. In the future, we will explore the potential of our model in other artificial intelligence areas, e.g., distant supervised relation extraction.

6. Acknowledge

This work is supported by the NSFC program (No. 61772151), and National Key R&D Program of China (No.2018YFC0830804).

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] Hannah Bast, Florian Bäurle, Björn Buchhold, and Elmar Haußmann. Easy access to the freebase dataset. In *WWW*, pages 95–98, 2014.

[3] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[4] Hung-Chen Chen, Zi-Yuan Chen, Sin-Yi Huang, Lun-Wei Ku, Yu-Shian Chiu, and Wei-Jen Yang. Relation extraction in knowledge base question answering: From general-domain to the catering industry. In *HCI*, pages 26–41, 2018.

[5] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *AAAI*, pages 301–306, 2011.

[6] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.

[7] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. *Machine Learning*, 94(2):233–259, 2014.

[8] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. In *SIGKDD*, pages 1156–1165, 2014.

[9] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[10] Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. Composing relationships with translations. In *EMNLP*, pages 286–290, 2015.

[11] Matt Gardner and Tom M. Mitchell. Efficient and expressive knowledge base completion using subgraph feature extraction. In *EMNLP*, pages 1488–1498, 2015.

[12] Matt Gardner, Partha Pratim Talukdar, Jayant Krishnamurthy, and Tom M. Mitchell. Incorporating vector space similarity in random walk inference over knowledge bases. In *EMNLP*, pages 397–406, 2014.

[13] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*, pages 1811–1818, 2018.

[14] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*, 2015.

Table 6: The weight assignment of different relation paths in FB15K. The order of the columns indicates the ranking of the weights of path attention (descending order), and the color depth of background in each path indicates the importance of relation attention (the deeper the more important).

Rank	Multi-hop Path
0	/location/location/people_born_here
1	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /location/location/people_born_here
2	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /people/person/place_of_birth^{-1}
3	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /people/person/places_lived./people/place_lived/location^{-1}
4	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor → /location/hud_county_place/place → /location/location/people_born_here
5	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor → /base/biblioness/bibs_location/country → /people/person/nationality^{-1}

Table 7: The visualization result of /location/country/languages_spoken.

Rank	Multi-hop Path
0	/location/country/languages_spoken
1	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /location/country/languages_spoken
2	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /language/human_language/countries_spoken_in
3	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor → /location/country/languages_spoken^{-1} → /location/location/adjoin_s./location/adjoining_relationship/adjoins
4	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /location/country/official_language
5	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor → /base/locations/continents/countries_within^{-1} → /language/human_language/region^{-1}

[15] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *ICLR* 2019.

[16] Abebaw Tadesse and Joseph B. Collins. Adversarial training on word-char embedding. In *AAAI-FSS*, pages 24–27, 2018.

[17] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, pages 2071–2080, 2016.

[18] Michihiro Yasunaga, Jungo Kasai, and Dragomir R. Radev. Robust multilingual part-of-speech tagging via adversarial training. In *NAACL-HLT*, pages 976–986, 2018.

[19] Pengda Qin, Weiran Xu, and William Yang Wang. DSGAN: generative adversarial training for distant supervision relation extraction. In *ACL*, pages 496–505, 2018.

[20] Xu Han, Zhiyuan Liu, and Maosong Sun. Denoising distant supervision for relation extraction via instance-level adversarial training. *CoRR* abs/1805.10959, 2018.

[21] Kentaro Inui, Ran Tian, and Ryo Takahashi. Interpretable and compositional relation learning by joint training with an autoencoder. In *ACL*, pages 2148–2159, 2018.

[22] Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, 2010.

[23] Ni Lao, Einat Minkov, and William W. Cohen. Learning relational features with backward random walks. In *ACL*, pages 666–675, 2015.

[24] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. In *EMNLP*, pages 705–714, 2015.

[25] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187, 2015.

[26] Aleix M. Martínez and Avinash C. Kak. Pca versus lda. *IEEE TPAMI*, 23(2):228–233, 2002.

[27] Sahisnu Mazumder and Bing Liu. Context-aware path ranking for knowledge base completion. In *IJCAI*, pages 1195–1201, 2017.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119. 2013.

[29] Kristina Toutanova, and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *3rd Workshop on CVSMC*, pages 57–66. 2015.

[30] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[31] Fedor Nikolaev, Alexander Kotov, and Nikita Zhiltsov. Parameterized fielded term dependence models for ad-hoc entity retrieval from knowledge graph. In *SIGIR*, pages 435–444, 2016.

[32] Marzia Polito and Pietro Perona. Grouping and dimensionality reduction by locally linear embedding. In *NIPS*, pages 1255–1262, 2001.

[33] Yelong Shen, Po-Sen Huang, Ming-Wei Chang, and Jianfeng Gao. Traversing knowledge graph in vector space without symbolic space guidance. *CoRR*, abs/1611.04642, 2016.

[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NIPS*, pages 2440–2448, 2015.

[35] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *CoRR* abs/1701.04862, 2017.

[36] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11:3371–3408, 2010.

[37] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–

Table 8: The visualization result of /people/person/education./education/education/institution.

Rank	Multi-hop Path
0	/people/person/education./education/education/institution
1	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /people/person/education./education/education/institution
2	/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency → /education/educational_institution/students_graduates./education/education/student^{-1}
3	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor → /people/person/education./education/education/institution → /education/educational_institution_campus/educational_institution
4	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor → /people/person/education./education/education/institution → /education/educational_institution/campuses^{-1}
5	/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor → /people/person/education./education/education/institution → /dataworld/gardening_hint/split_to^{-1}

1119, 2014.

[38] Zhuoyu Wei, Jun Zhao, and Kang Liu. Mining inference formulas by goal-directed random walks. In *EMNLP*, pages 1379–1388, 2016.

[39] Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard H. Hovy. An interpretable knowledge transfer model for knowledge base completion. In *ACL*, pages 950–962, 2017.

[40] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *NAACL-HLT*, pages 1480–1489, 2016.

[41] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In *AAAI*, pages 3060–3067, 2019.

[42] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen and Dinh Q. Phung. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *NAACL*, pages 327–333, 2018.

[43] Masashi Yoshikawa, Koji Mineshima, Hiroshi Noji and Daisuke Bekki. Combining Axiom Injection and Knowledge Base Completion for Efficient Natural Language Inference. In *AAAI*, pages 7410–7417, 2019.

[44] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen and Dinh Q. Phung. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. In *NAACL-HLT*, pages 2180–2189, 2019.

[45] Zhao Zhang, Fuzhen Zhuang, Zheng-Yu Niu, Deqing Wang and Qing He. MultiE: Multi-Task Embedding for Knowledge Base Completion. In *CIKM*, pages 1715–1718, 2018.

[46] Timothée Lacroix, Nicolas Usunier and Guillaume Obozinski. Canonical Tensor Decomposition for Knowledge Base Completion. In *ICML*, pages 2869–2878, 2018.

[47] Alexandros Komninos and Suresh Manandhar. Feature-Rich Networks for Knowledge Base Completion. In *ACL*, pages 324–329, 2017.