World Wide Web (2021) 24:419–439
https://doi.org/10.1007/s11280-020-00847-2

Open-world knowledge graph completion with multiple interaction attention

Lei Niu[1] · Chenpeng Fu[1] · Qiang Yang[2] · Zhixu Li[1,3] · Zhigang Chen[4] · Qingsheng Liu[5] · Kai Zheng[6,7]

Received: 19 March 2020 / Revised: 14 September 2020 / Accepted: 21 September 2020 /
Published online: 28 October 2020

Abstract

Knowledge Graph Completion (KGC) aims at complementing missing relationships between entities in a Knowledge Graph (KG). While closed-world KGC approaches utilizing the knowledge within KG could only complement very limited number of missing relations, more and more approaches tend to get knowledge from open-world resources such as online encyclopedias and newswire corpus. For instance, a recent proposed open-world KGC model called ConMask learns embeddings of the entity's name and parts of its text-description to connect unseen entities to the KGs. However, this model does not make full use of the rich feature information in the text descriptions, besides, the proposed relationship-dependent content masking method may easily miss to find the target-words. In this paper, we propose to use a Multiple Interaction Attention (MIA) mechanism to model the interactions between the head entity description, head entity name, the relationship name, and the candidate tail entity descriptions, to form the enriched representations. In addition, we try to use the additional textual features of head entity descriptions to enhance the head entity representation and apply the attention mechanism between candidate tail entities to enhance the representation of them. Besides, we try different scoring functions to increase the convergence of the model. Our empirical study conducted on three real-world data collections shows that our approach achieves significant improvements comparing to state-of-the-art KGC methods.

Keywords Knowledge graph completion · Attention · Open-world

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering 2019*
Guest Editors: Reynold Cheng, Nikos Mamoulis, and Xin Huang

✉ Kai Zheng
zhengkai@uestc.edu.cn

Extended author information available on the last page of the article.

1 Introduction

Knowledge Graph (KG) is a kind of large-scale structured semantic network whose nodes represent entities and edges represent relations between entities [9]. The rise of KGs in the past years has made great contributions to the success of many applications such as entity linking [10], recommendation [36] and question answering [25].

As more and more entities involved in a KG, a large portion of relations between entities might be missing. To deal with the case, the task of Knowledge Graph Completion (KGC) is proposed, aiming at complementing missing relation between entities in KGs. In the past years, a lot of attention has been paid on this topic, which can be roughly divided into closed-world KGC approaches and open-world KGC approaches.

The closed-world KGC approaches tend to utilize the knowledge within KGs. The most active closed-world KGC methods are based on the knowledge graph embedding models such as TransE [4] and its variances [14, 21, 38]. By encoding the entities and relations between entities in KGs into a continuous low-dimensional embedding vectors space, we could do some inference to identify some hidden relations between entities. However, closed-world KGC approaches could only complement very limited number of relation, i.e., usually lead to a low recall. On the other hand, some work tends to get knowledge from open-world resources such as online encyclopedias and newswire corpus. For instance, Description-Embodied Knowledge Representation Learning (DKRL) [39] proposes to learn the representations of entities from not only TransE, but also the description of the entities in online-encyclopedias. To achieve this, DKRL adopts to do a joint training for graph-based embeddings and description-based embeddings. They use continuous bag-of-words and deep convolutional neural network models to encode semantics of entity descriptions. However, it does not take into account that various relationships focus on different parts of the entity description, and not all information provided in its entity description is useful to predict linked entities of a given specific relationship.

A recent proposed open-world KGC model called ConMask [30] learns embeddings of an entity's name and parts of its text-description to connect unseen entities to the KGs. As illustrated in Figure 1, it first uses a so-called relationship-dependent content masking approach to select the words related to the given relationship in the relevant text description, which could effectively mitigate the presence of noisy text descriptions. Next, it trains a Fully Convolutional Neural network (FCN) to extract the word-based target entity

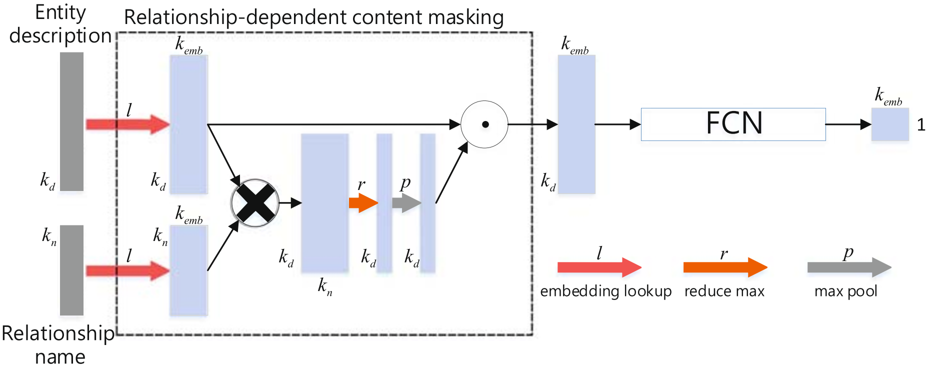


Figure 1 Framework of Conmask model, where k_d is the length of the entity description, k_n is the length of the relationship name and k_{emb} is the word embedding size

embeddings from relevant text descriptions. Finally, the extracted word-based target entity embeddings and other textual features (entity names) are compared with the existing target candidate entities in the KG to resolve a ranked list of target candidate entities.

However, there are at least two **weaknesses** with the ConMask model. Firstly, the information of entity descriptions is not fully used. Now only the pre-trained word embeddings, is used for the representation of words in the text descriptions, some potential semantic and statistic information might be missing. Secondly, the proposed relationship-dependent content masking model could only find possible target-words that appear in the fixed-size content masking window after the indicator word, without considering that the situation that the target-words could also appear in front of the indicator word. Besides, it is difficult to determine a proper size for the content masking window.

In our previous work [7], we propose a novel open-world KGC approach based on the same input resources with ConMask, i.e., entity names, relationship names, and entity descriptions. But different from ConMask which only uses the entity descriptions in a very simple way, we propose to use attention mechanisms to fully capture the important information generated from the multiple interactions between entity names, relationship names and entity descriptions. More specifically, the multiple interactions involved in the model include: (1) The interaction between the head entity name, the relationship name, and the head entity description. (2) The interaction between the head entity, the relationship, and the candidate tail entity. (3) The interaction between the description of the head entity and the candidate tail entity descriptions In this way, our Multiple Interaction Attention (MIA) model could not only flexibly select relevant parts of the entity description according to different relationships, but also better aware of the relevant part in the head entity description and obtain the head-aware representation of the candidate tail entity description.

In this paper, We introduce the interaction between multiple candidate tail entity descriptions so that our model can explore the hidden relationships among multiple tail entity descriptions and use such relationships to form the interactions between them and enhance their representation. In addition, in the final layer of the model, we also design several scoring functions to compare the convergence ability of the model under different functions and to enhance the effectiveness of our model.

Besides, to make effective use of the rich information in the entity descriptions, our model encodes the head entity description, head entity name, the relationship name, and the candidate tail entity descriptions into word representations which are enhanced by additional Part-Of-Speech (POS) tags, Named-Entity-Recognition (NER) tags and handcrafted textual features.

To summarize, our contributions in this paper can be summarized as follows:

- We propose to use attention mechanism to simulate the interaction between the head entity name, the relationship name and the entity descriptions, such that we could dynamically select the most related information from the head entity description and the candidate tail entity descriptions according to different relations.
- We use the attention mechanism to align relevant parts between the head entity description and the candidate tail entity descriptions, such that we could enrich the representation of the candidate tail entity description.
- We use the attention mechanism to obtain hidden relationships between multiple candidate tail entities and use them to enhance the representation of them.
- We also propose to make effective use of the rich information in the entity descriptions with some additional important features.

Our empirical study conducted on three real-world data collections shows that our approach achieves significant improvements on open-world KGC compared with state-of-the-art methods.

The rest of this paper is organized as follows: We cover the related work in Section 2, and then define our problem and introduce the framework of our approach in Section 3. After giving details of MIA model in Section 4, we report the empirical study in Section 5. We finally conclude the paper in Section 6.

2 Related work

Knowledge graph completion (KGC) aims at completing the missing relation between entities in given KG (or KGs). So far, a lot of attention has been paid on this topic, which can be roughly divided into closed-world KGC approaches and open-world KGC approaches.

2.1 The closed-world knowledge graph completion

The closed-world KGC approaches tend to utilize the knowledge within KGs. The most active closed-world KGC methods are based on the knowledge graph embedding models such as TransE [4] and its variances [1, 14, 16, 21, 23, 34, 38]. By encoding the entities and relations between entities in the KG into a continuous low-dimensional embedding vectors space, we could do some inference to identify some hidden relations between entities.

For a given triple *(head entity, relationship, tail entity)*, also denoted as (h, r, t), the typical embedding-based KGC model TransE [4] assumes the energy function is defined as

$$E(h, r, t) = \| \mathbf{h} + \mathbf{r} - \mathbf{t} \|, \tag{1}$$

which indicates that the tail embedding $\mathbf{t}$ should be the closeness neighbour of $\mathbf{h} + \mathbf{r}$, where $\mathbf{h}$, $\mathbf{r}$ are embeddings of head entity and relationship respectively. There are also many models that introduce more relationship-dependent parameters. TransR [21], $\mathbf{hM}_r + \mathbf{r} = \mathbf{tM}_r$ where $\mathbf{M}_r$ is a relationship-dependent entity embedding transformation. TransR [21] models entities and relations in distinct semantic space (entity space and relation spaces) and performs translation in relation space when learning embeddings. PTransE [20] maintain a simple translation function and proposes a multiple-step relation path-based representation learning model. SimplE [16] and TuckER [1] employ tensor decomposition to train and obtain representations of head entities, tail entities and relations. RotatE [34] designs the model for three special relationships: symmetric/antisymmetric, inversion, and composition. Liu et al. [22, 24] focus on the optimal social trust path selection problem in complex social networks.

Unlike topology-based models that have been extensively studied, there are several methods that use textual information for KGC. For instance, the Neural Tensor Networks (NTN) model [31] initializes the representation of the entity by using the average word embedding in entity name, and allow sharing of textual information located in similar entity names. Zhang et al. [42] represents entities with entity names or the average of word embeddings in descriptions. Jointly [40] first uses the weighted sum combination topology-embeddings and text-embeddings, and then calculates the L_n distance between the translated head and tail entities. However, closed-world KGC approaches could only complement very limited number of relations, i.e., usually lead to a low recall.

2.2 The open-world knowledge graph completion

More recent work tends to get knowledge from open-world resources such as online encyclopedias and newswire corpus. In traditional research, such external knowledge is often used to explore new relationships in KGs, which is often called relation extraction [26]. The common applications tend to use neural networks such as CNN [40] or abstract meaning representations (AMRs) [13]. When it comes to KGC, DKRL [39] uses a joint training of graph-based embeddings and description-based embeddings. They use continuous bag-of-words and deep convolutional neural network models to encode semantics of entity descriptions. It can directly build representations from the description of the novel entities. A recent work proposes ConMask [30] model, which is a text-focused approach that could reduce irrelevant and noisy words by selecting words associated with relationships in the given entity description, and then fuse the relevant text through fully convolutional neural networks (FCN) to extract the word-based entity embedding and combined with background representations of other textual features (entity names) to connect unseen entities to the KG.

2.3 Text-focused studies in machine reading comprehension

ConMask is designed to get information from the description text to help with KGC. The problem of extracting required information from texts for a given question is well studied in the field of machine reading comprehension [15, 32, 35, 43, 44]. There are two main research directions of machine reading comprehension: generative reading comprehension and multiple-choice reading comprehension. The goal of the former is to extract the answer of the question from a given text and question where the dominant trend is a variety of attention-based interactions between text and question. For example, Kadlec et al. propose a method called "Attention Sum Reader" [15], which directly extracts the answer from the text using attention mechanism. Recently, Cui et al. propose an attention-over-attention(AoA Reader) neural networks for reading comprehension, which reduces the computational complexity of the model. The performance is improved further due to the usage of document-level attention. When it comes to multiple-choice reading comprehension, researchers try to introduce information in candidate answers into the model. Trischler et al. [35] propose a parallel-hierarchical neural model that matches the texts, questions and candidate answers from from word level to sentence level. However the model must be trained with the training wheel [32] to converge. In recent work, Zhu et al. [44] propose a model which uses hierarchical attention flow to enhance the interaction between the candidate answers and the text. The experimental results on their datasets show that their method significantly outperforms various state-of-the-art systems.

3 Problem definition and the framework

We formally define the Knowledge Graph Completion (KGC) task below:

Definition 1 (**Knowledge Graph Completion (KGC)**) Given a Knowledge Graph $\mathcal{KG}$ with a set of incomplete relation triples in the form of $(h, r, ?)$, where h denotes the head entity, r denotes the relation, and the ? is the missing tail entity t, the task of Knowledge Graph Completion (KGC) is to find t for each incomplete relation triple to consist a complete one (h, r, t).

To illustrate how our Multiple Interaction Attention (MIA) model works on open-world KGC task, several examples are given in Figure 2. For a given partial triple ($Donald\_Trump$, mother, ?), if a human reader were asked to determine from the head entity description and some candidate tail entity descriptions, "Who is the mother of US President Donald Trump?", then human reader will first look for contextual clues such as mother, parent or family-related information from the description of the head entity "$Donald\_Trump$". Here, the human reader has located the sentence "His parents were ... and Scottish-born housewife Mary Anne MacLeod" in the head entity description. So, the human reader may infer that Donald Trump's mother is a Scottish-born housewife Mary Anne MacLeod. After that, the human reader locates the description of the candidate tail entity "$Mary\_Anne\_MacLeod\_Trump$" from the candidate tail entity descriptions. In the description of "$Mary\_Anne\_MacLeod\_Trump$", the human reader will be pleasantly surprised to find "Mary Anne Trump was the mother of Donald Trump, the 45th president of the United States" and "Born in the Outer Hebrides of Scotland". Therefore, the human reader can more accurately reason that "$Mary\_Anne\_MacLeod\_Trump$" is the correct tail entity of the partial triple ($Donald\_Trump$, mother, ?).

We split the above reasoning process into three steps below:

1. Locating task-related information in the head entity description and the candidate tail entity descriptions, respectively;
2. Extracting the context information of the related text in the head entity description and the candidate tail entity descriptions;
3. Matching the head entity description and candidate tail entity descriptions respective relevant text context information to determine the correct tail entity.

Correspondingly, the MIA model is designed to simulate this process, which is mainly composed of three components below:

1. Multiple Interaction Attention, which highlights task-related words;

<table>
<tr><td>Partial triple:
(<i>Donald_Trump,</i> birth_ place, ?)</td><td>Partial triple:
(<i>Donald_Trump,</i> mother, ?)</td><td>Partial triple:
<i>(Fred_Trump,</i> spouse , ?)</td></tr>
<tr><td colspan="2">Head entity description:
<i>Donald_Trump</i>: Donald John Trump is the 45th and current president of the United States.... Trump was born and raised in the New York City borough of Queens and received an economics degree from the Wharton School. . . . His parents were Frederick Christ Trump, a real estate developer, and Scottish-born housewife Mary Anne MacLeod. ... Trump had graduated first in his class at Wharton. ...</td><td>Head entity description:
<i>Fred_Trump</i>: Frederick Christ Trump ... and the father of Donald Trump, the 45th president of the United States,...Trump met his future wife Mary Anne MacLeod , an immigrant from Glasgow, Scotland, ...</td></tr>
<tr><td>Question:
donald trump birth place ?</td><td>Question:
donald trump mother ?</td><td>Question:
frederick christ trump spouse ?</td></tr>
<tr><td>Candidate tail entity descriptions:
<i>Washington,_ D.C</i>: Washington, D.C., formally the District of Columbia and commonly referred to as Washington or D.C., is the capital of the United States. Founded after the American ...
...
<i>Beijing</i>: Beijing, alternately romanized as Peking, is the capital of the People's Republic of China, the world's third most populous city proper, ...
...
<i>New_York_City</i>: The City of New York, usually called either New York City or simply New York (NY), is the most populous city in the United States. ... five boroughs – Brooklyn, Queens, ...
...</td><td>Candidate tail entity descriptions:
<i>Ann_Dunham</i>: Stanley Ann Dunham was an American anthropologist...She was the mother of Barack Obama, the 44th President of the United States. ...
...
<i>Mary_Anne_MacLeod_Trump</i>: Mary Anne Trump was the mother of Donald Trump, the 45th president of the United States , . . . Born in the Outer Hebrides of Scotland, ... Mary Anne's husband Fred Trump died at age 93 in June 1999. ...
...
<i>Dorothy_Howell_Rodham</i>: Dorothy Emma Rodham was an American homemaker ... and 2016 Democratic presidential nominee Hillary Clinton. ...
...</td><td>Candidate tail entity descriptions:
<i>Michelle _ Obama</i> : Michelle LaVaughn Obama is an American lawyer, ...She is married to the 44th U.S. president, Barack Obama, ...
...
<i>Mary_Anne_MacLeod_Trump</i>: Mary Anne Trump was the mother of Donald Trump, the 45th president of the United States , . . . Born in the Outer Hebrides of Scotland, ... Mary Anne's husband Fred Trump died at age 93 in June 1999 . ...
...
<i>Hillary_ Clinton</i>: Hillary Diane Rodham Clinton is an American politician ... she moved to Arkansas and married future president Bill Clinton in 1975; ...
...</td></tr>
<tr><td>Corrent tail entity:
<i>New_York_City</i></td><td>Corrent tail entity:
<i>Mary_Anne_MacLeod_Trump</i></td><td>Corrent tail entity:
<i>Mary_Anne_MacLeod_Trump</i></td></tr>
</table>

Figure 2 Open-world KGC examples with our MIA model

2. Text Context Encoder, which encodes context information in the relevant text;
3. Matching Prediction, which chooses a correct tail entity by matching the context information in the relevant text to calculate the similarity score between the head entity description and the candidate tail entity descriptions.

Note that we consider that the head entity, relationship, and tail entity usually appear in a snippet of the text description at the same time, so we combine the head entity name with the relationship name into a question as an input to our model to help the model locate task-related information more effectively.

4 The MIA model

The MIA model first encodes the head entity description, question, and candidate tail entity descriptions into a word representation and enhances it by appending some other features. Then, it emphasizes and organizes relevant information by using a word-level attention mechanism to simulate the interaction between the head entity description, question and candidate tail entity descriptions. Afterwards, MIA uses Bidirectional Long Short-Term Memory network (Bi-LSTM) to encoded context information in the relevant text. After that, it uses a word-level attention mechanism between multiple candidates to enhance the representation of tail entities. Finally, through a matching prediction, it compares the representation extracted to the head entity description with the representation of each candidate tail entity description to resolve a ranked list of candidate tail entities. The architecture of MIA model is also illustrated in Figure 3.

In the following of this section, we describe the details of the MIA model component by component. Throughout this section, we will use D_h for representing the head entity description, Q_r for representation question consisting of the head entity name and the relationship name, C_t for the candidate tail entity descriptions, and C_{t_i} for the description of the i-th candidate in the candidates set for the same question to be completed. Note that since

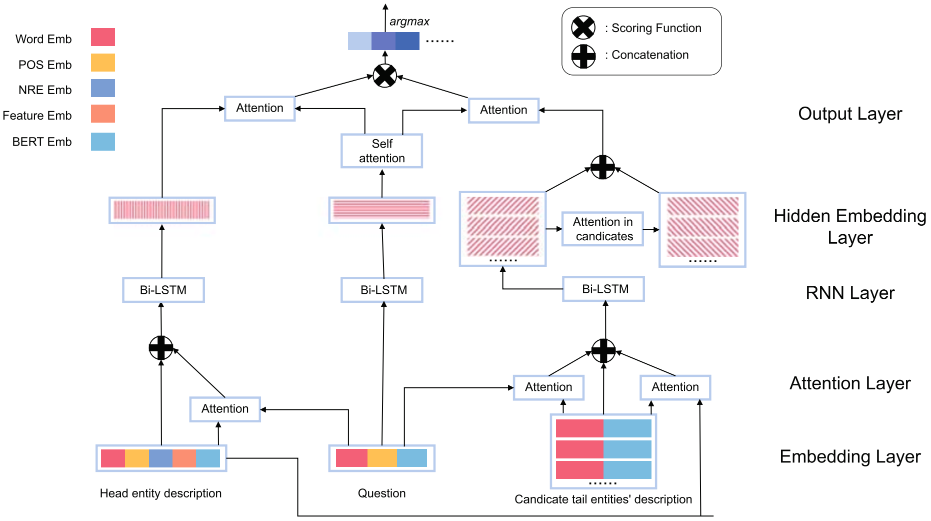


Figure 3 Main neural architecture of the Multiple Interaction Attention (MIA) model

the description of the operations for each candidate tail entity are the same, for the sake of simplicity, we only take one of the candidate tail entity descriptions for illustration.

4.1 Input word representation

We transform each word in the head entity description, question, and candidate tail entity description into continuous representations. In this paper, each training example entered during training contains a head entity description $\{w_i^{D_h}\}_{i=1}^{|D_h|}$, a question $\{w_j^{Q_r}\}_{j=1}^{|Q_r|}$, a candidate tail entity description $\{w_n^{C_t}\}_{n=1}^{|C_t|}$ and a label $y^* \in \{0, 1\}$, where $|D_h|$, $|Q_r|$, and $|C_t|$ are the length of the head entity description, question, and candidate tail entity description.

Here, we take the input representation of the i-th word $w_i^{D_h}$ in the given head entity description as an example, which is the concatenation of several components:

- **Word Embedding** : We use the publicly available pre-trained GloVe [27] embedding $\mathbf{E}_{w_i^{D_h}}^{word}$.
- **Part-Of-Speech (POS) Embedding** : We use spaCy[1] for part-of-speech tagging $\mathbf{E}_{w_i^{D_h}}^{pos}$. Similar to traditional word embeddings, we assign different trainable vectors for each part-of-speech tag.
- **Named-Entity-Recognition (NER) Embedding** : Like POS, we use spaCy for named entity recognition $\mathbf{E}_{w_i^{D_h}}^{ner}$.
- **Handcrafted Features Embedding** : We use term frequency and co-occurrence feature as handcrafted features $\mathbf{E}_{w_i^{D_h}}^{feat}$. The term frequency is calculated based on English Wikipedia. In the binary co-occurrence feature, it is true when $w_i^{D_h}$ appears in the question $\{w_j^{Q_r}\}_{j=1}^{|Q_r|}$ or candidate tail entity description $\{w_n^{C_t}\}_{n=1}^{|C_t|}$.
- **BERT Embedding** : **BERT** [6] is a NLP model developed by Google for pre-training language representations. It leverages an enormous amount of plain text data publicly available on the web and is trained in an unsupervised manner. We use BERT-as-service[2] to get the word representation $\mathbf{E}_{w_i^{D_h}}^{bert}$

We concatenate five embedding components to form the final input representations for the word $w_i^{D_h}$, namely $\mathbf{E}_{w_i^{D_h}}$.

$$\mathbf{E}_{w_i^{D_h}} = \left[\mathbf{E}_{w_i^{D_h}}^{word}; \mathbf{E}_{w_i^{D_h}}^{pos}; \mathbf{E}_{w_i^{D_h}}^{ner}; \mathbf{E}_{w_i^{D_h}}^{feat}; \mathbf{E}_{w_i^{D_h}}^{bert}\right] \tag{2}$$

In the same way, we concatenate Word Embedding $\mathbf{E}_{w_j^{Q_r}}^{word}$, POS Embedding $\mathbf{E}_{w_j^{Q_r}}^{pos}$ and BERT Embedding $\mathbf{E}_{w_j^{Q_r}}^{bert}$ to get the input word representation $\mathbf{E}_{w_j^{Q_r}}$ of a word $w_j^{Q_r}$ in a given question.

$$\mathbf{E}_{w_j^{Q_r}} = \left[\mathbf{E}_{w_j^{Q_r}}^{word}; \mathbf{E}_{w_j^{Q_r}}^{pos}; \mathbf{E}_{w_j^{Q_r}}^{bert}\right] \tag{3}$$

[1] https://github.com/explosion/spaCy

[2] https://github.com/hanxiao/bert-as-service

The input representation of the $w_n^{C_{t_i}}$ for a given candidate tail entity description C_{t_i} contains Word Embedding $\mathbf{E}^{word}_{w_n^{C_{t_i}}}$ and BERT Embedding $\mathbf{E}^{bert}_{w_n^{C_{t_i}}}$.

$$\mathbf{E}_{w_n^{C_{t_i}}} = \left[\mathbf{E}^{word}_{w_n^{C_{t_i}}}; \mathbf{E}^{bert}_{w_n^{C_{t_i}}}\right] \tag{4}$$

4.2 Multiple interaction attention

In our model, we use the interaction between the head entity, the question, and the candidate tail entity to emphasize and organize relevant information accordingly. We exploit the same word-level sequence alignment attention mechanism for each interaction. In this section, we first describe the Word-level Sequence Alignment (WSA) attention mechanism in detail and then explain various interactions.

Word-level sequence alignment attention mechanism Following [5, 18, 37] and other recent works, given two inputs $\mathbf{X}$ and $\mathbf{Y} = \{\mathbf{Y}_i\}_{i=1}^m$, let's define the attention function:

$$att(\mathbf{X},\ \{\mathbf{Y}_i\}_{i=1}^m) = \sum_{i=1}^m a_{xy_i}\mathbf{Y}_i \tag{5}$$

$$a_{xy_i} = \text{softmax}(\alpha(\mathbf{WX})^T\alpha(\mathbf{WY}_i)) \tag{6}$$

where the attention score a_{xy_i} captures the similarity between $\mathbf{X}$ and each words $\mathbf{Y}_i$, $\mathbf{W}$ is a matrix, and $\alpha(\cdot)$ is a activation function with ReLU nonlinearity.

Question-aware head entity description WSA attention Note that words in the head entity description are not equally important, and the importance of them changes in tune with the different questions. Just like people find relevant answers from a given passage based on the question, people can always give more attention to the words that are most relevant to the question. Therefore, we can get the question-aware representation $\mathbf{E}^{q_r}_{w_i^{D_h}}$ of the word $w_i^{D_h}$ in the head entity description according to the question:

$$\mathbf{E}^{q_r}_{w_i^{D_h}} = att(\mathbf{E}^{word}_{w_i^{D_h}},\ \{\mathbf{E}^{word}_{w_j^{Q_r}}\}_{j=1}^{|Q_r|}) \tag{7}$$

Question-aware candidate tail entity description WSA attention In a similar way, we use question information as the key to extracting important information from candidate tail entity description. For each candidate tail entity description C_{t_i}, we get the question-aware representation $\mathbf{E}^{q_r}_{w_n^{C_{t_i}}}$ of the word $w_n^{C_{t_i}}$ in the candidate tail entity description:

$$\mathbf{E}^{q_r}_{w_n^{C_{t_i}}} = att(\mathbf{E}^{word}_{w_n^{C_{t_i}}},\ \{\mathbf{E}^{word}_{w_j^{Q_r}}\}_{j=1}^{|Q_r|}) \tag{8}$$

Head-aware candidate tail entity description WSA attention We find when those entities have relationships, they usually mention to each other in each other's descriptions. In order to adequately leverage the information in the head entity description, we align the candidate tail entity description with the head entity descriptions. In details, we embed the information of the head entity description into the candidate tail entity description representation so that we can better align and aware the relevant parts of the head entity description. Thereby the word $w_n^{C_{t_i}}$ in the candidate tail entity description C_{t_i} can obtain the aware representation of the head entity description with the following equation:

$$\mathbf{E}^{d_h}_{w_n^{C_{t_i}}} = att\left(\mathbf{E}^{word}_{w_n^{C_{t_i}}},\ \left\{\mathbf{E}^{word}_{w_i^{D_h}}\right\}_{i=1}^{|D_h|}\right) \tag{9}$$

4.3 Text context encoder

The third component of the model is the Recurrent Neural Network (RNN) layer which uses a Bidirectional Long Short-Term Memory network (Bi-LSTM) [8, 29] to model the contextual information. In addition, after RNN layer, an attention mechanism between multiple candidate tail entity descriptions is applied to obtain the enhanced representation of these descriptions.

In order to learn long-term dependencies [2, 11, 12] in RNN, Long Short-Term Memory network (LSTM) was proposed by [12]. The Bi-LSTM consists of two independent LSTMs, the forward LSTM and the backward LSTM. By using three separate Bi-LSTMs, we encode the head entity description, question and candidate tail entity description as follows:

$$\mathbf{B}^{D_h} = \text{Bi-LSTM}\left(\left\{\left[\mathbf{E}_{w_i^{D_h}}; \mathbf{E}^{q_r}_{w_i^{D_h}}\right]\right\}_{i=1}^{|D_h|}\right) \tag{10}$$

$$\mathbf{B}^{Q_r} = \text{Bi-LSTM}\left(\left\{\mathbf{E}_{w_j^{Q_r}}\right\}_{j=1}^{|Q_r|}\right) \tag{11}$$

$$\widehat{\mathbf{B}}^{C_{t_i}} = \text{Bi-LSTM}\left(\left\{\left[\mathbf{E}_{w_n^{C_{t_i}}}; \mathbf{E}^{q_r}_{w_n^{C_t}}; \mathbf{E}^{d_h}_{w_n^{C_{t_i}}}\right]\right\}_{n=1}^{|C_{t_i}|}\right) \tag{12}$$

Attention between multiple candidate tail entity descriptions The candidate tail entity representation $\widehat{\mathbf{B}}^{C_{t_i}}$ is generated by the WSA attention which is aware of the question and the head entity description. However such representation is independent of other candidates and does not encode the hidden relationship information between the candidates. Inspired by Zhu et al. [44], there are also hidden relationships between candidate tail entities that are helpful in finding the right answer. For example, fragments of the correct candidate tail entity description may appear frequently in the descriptions of other candidate tail entities. So we design a new attention layer to explore the hidden relationships between candidates and obtain new candidate tail entity representation $\widehat{\widehat{\mathbf{B}}}^{C_{t_i}}$. We train a matrix W_{cc} to calculate the impact factors between the candidates, which are used as weights in the subsequent aggregation process.

$$A_{i,j} = \widehat{\mathbf{B}}^{C_{t_i}} W_{cc} \widehat{\mathbf{B}}^{C_{t_j}} \tag{13}$$

$$w_{i,j} = \frac{exp(A_{i,j})}{\sum_{k=0}^{m} exp(A_{i,k})} \tag{14}$$

where m is the count of candidate tail entities.

$$\widehat{\widehat{\mathbf{B}}}^{C_{t_i}} = \sum_{k=0}^{m} w_{i,k}\widehat{\mathbf{B}}^{C_{t_k}} \tag{15}$$

Then, we model the candidate correlations with difference $\widehat{\mathbf{B}}^{C_{t_i}} - \widehat{\widehat{\mathbf{B}}}^{C_{t_i}}$, which is inspired by Chen et al. [28]. At last, we concatenate the difference to the independent candidate representation for enhancement.

$$\mathbf{B}^{C_{t_i}} = [\widehat{\mathbf{B}}^{C_{t_i}}; \widehat{\mathbf{B}}^{C_{t_i}} - \widehat{\widehat{\mathbf{B}}}^{C_{t_i}}] \tag{16}$$

4.4 Matching prediction

We use the self-attention [41] to summarize the question sequence representation $\mathbf{B}^{Q_r}$ into the final question representation $\mathbf{R}_{Q_r}$. The definition of the self-attention function is as follows:

$$att_{self}(\{\mathbf{X}_i\}_{i=1}^{m}) = \sum_{i=1}^{m} a_i \mathbf{X}_i \tag{17}$$

$$a_i = \text{softmax}\left(\mathbf{W}_{self}^{T}\mathbf{X}_i\right) \tag{18}$$

where the attention score a_i indicates the importance of $\mathbf{X}_i$ in $\{\mathbf{X}_i\}_{i=1}^{m}$.

According to the question representation $\mathbf{R}_{Q_r} = att_{self}(\{\mathbf{B}_j^{Q_r}\}_{j=1}^{|Q_r|})$, we can get the head entity description representation $\mathbf{R}_{D_h} = att(\mathbf{R}_{Q_r}, \{\mathbf{B}_i^{D_h}\}_{i=1}^{|D_h|})$, and the i-th candidate tail entity description representation $\mathbf{R}_{C_{t_i}} = att(\mathbf{R}_{Q_r}, \{\mathbf{B}_n^{C_{t_i}}\}_{n=1}^{|C_{t_i}|})$.

Instead of simply multiplying two vectors to get the score, we try a variety of functions to get the final score, mainly including the following three functions:

$$Score_{linear}(\mathbf{R}_{D_h}, \mathbf{R}_{C_{t_i}}) = (\mathbf{R}_{D_h})^T\, W_{S_1}\, \mathbf{R}_{C_{t_i}} \tag{19}$$

$$Score_{bilinear}(\mathbf{R}_{D_h}, \mathbf{R}_{C_{t_i}}) = (\mathbf{R}_{D_h})^T\, W_{S_2}\, ((\mathbf{R}_{C_{t_i}})^T\, W_{S_2})^T \tag{20}$$

$$Score_{trilinear}(\mathbf{R}_{D_h}, \mathbf{R}_{C_{t_i}}) = W_{S_3}[\mathbf{R}_{D_h}; \mathbf{R}_{C_{t_i}}; \mathbf{R}_{D_h}\mathbf{R}_{C_{t_i}}] \tag{21}$$

newwhere W_{S_1}, W_{S_2} and W_{S_3} are the transformation matrices that need to be trained. By score function, each candidate tail entity has its score s_i and we set the output of model y' as follows:

$$s_i = Score(\mathbf{R}_{D_h}, \mathbf{R}_{C_{t_i}}) \tag{22}$$

$$y' = softmax([s_1; s_2; \ldots; s_m]) \tag{23}$$

where s_i represents the probability that its corresponding candidate tail entity is correct and new$Score(\cdot)$ refers to one of them, $Score_{linear}$, $Score_{bilinear}$ or $Score_{trilinear}$ whose performance is illustrated in experiments.

To train our model, we use softmax cross entropy function as the loss function to minimize the gap between the prediction and the ground truth.

$$L = -\sum_{i=1}^{m} y_i log y_i' \tag{24}$$

where y is the one-hot encoding of the label of sample. y_i and y_i' represent i-th value of y and y'.

5 Experiments

5.1 Datasets

We use the following three public-accessed datasets for evaluating the performance of our approach in open-World knowledge graph completion. (1) FB15k dataset [4], a dataset extracted from a typical large-scale KG Freebase [3]. The dataset contains about 15,000 entities and 580,000 relational triples between entities and is often used to evaluate the effectiveness of closed-World knowledge graph completion models. (2) FB20k dataset [39] , the dataset is built upon the FB15k dataset, it first removed 47 entities from FB15K which have shorter than 3 words after preprocessed or even have no descriptions, and removed all triples containing these entities in FB15K, then by adding test triples with unseen entities, which are selected to have rich descriptions. (3) DBPedia50k dataset [30] for both open-world and closed-world KGC tasks, a dataset randomly sampled from a large-scale KG DBPedia [19]. It is worth mentioning that in order to evaluate the effectiveness of our model in open-World knowledge graph completion, we extract 2000 entities from the entity set and make sure that these 2000 entities do not appear in the training set when we divide the FB15k dataset. This represents that the test set and validation set contain 2000 entities not included in the training set. In addition, FB15k also removes these 47 entities and the triples that contain them. We denote the processed FB15K as FB15k$_{\text{open}}$. In addition, for FB20k and DBpedia50k, Shi et al. also used a similar approach to ensure that the test set and validation set contain entities not included in the training set [30]. We evaluate our approach on FB15k$_{\text{open}}$, FB20k and DBPedia50k. Statistics of datasets are shown in Table 1.

5.2 Experiment setting

Due to the lack of an open-world KGC task validation set on FB20k, we randomly sampled 10% of the test triples as a validation set.

Evaluation protocol We use the tail entity prediction on the test set for performance evaluation. For each test triple (h, r, t) with open-world head entity $h \in \mathbf{E}'$,where $\mathbf{E}'$ is an entity superset, we rank all known entities $t \in \mathbf{E}$ by use the KGC model to calculate the actual ranking score, where $\mathbf{E}$ is an entity set. We then use three measures as our evaluation metrics: (1) Mean Rank (MR): the averaged rank of correct tail entities; (2) HITS@K: the proportion of correct tail entities ranked in top k; (3) Mean Reciprocal Rank (MRR): mean reciprocal rank of correct tail entities.

Note that there may be multiple triples in the dataset that have the same head entity and relationship but different tail entities: (h, r, t_1), ..., (h, r, t_n). Following [4], when computing the Mean Reciprocal Rank (MRR), given a triple (h, r, t_i) only the reciprocal rank of t_i itself is evaluated (and not the best out of t_1, ..., t_i, ..., t_n, which would produce better results).

Table 1 Data set statistics

Dataset	Entities	Rel.	Train	Validation	Test
FB15k$_{\text{open}}$	14,904	1,341	470,313	48,991	60,377
FB20k	19,923	1,341	472,860	48,991	88,293
DBPedia50k	49,900	654	32,388	399	10,969

Table 2 Hyper-parameter settings

Symbol	Descriptions	Size
$\|D_h\|$	Head entity description max length	512
$\|C_t\|$	Candidate tail entity description max length	512
k	Word embedding size	200
pos	POS-tag embedding size	12
ner	NRE-tag embedding size	8
h	Bi-LSTM hidden size	96

This differs from ConMask's MRR evaluation method, which is the reason why result in Table 3 differs from [30] (see the asterisk (*) mark).

Note also that a filtering method called $target\ \ filtering$ is used in ConMask: When evaluating a test triple (h, r, t), only when a triple of the form $(?, r, t')$ exist in the training set, we treat the tail entity t' as a candidate tail entity, otherwise it is skipped. Therefore, we also use $target\ \ filtering$ when comparing with the Conmask model.

Parameter setting Following ConMask, we set the maximum head entity description length $|D_h| \leq 512$ and the maximum candidate tail entity description length $|C_t| \leq 512$. We apply the spaCy for tokenization, part-of-speech (POS), and named entity recognition (NER). The main hyper-parameters of our model are listed in Table 2. The word embeddings are initialized by the publicly available pre-trained 200-dimensional GloVe [27] embeddings. We use Adam [17] for parameter optimization, with initial learning rate 0.002. A mini-batch of 32 samples is used to update the model parameter per step. In order to prevent overfitting, we apply dropout [33] to input embeddings and Bi-LSTM's outputs with a drop rate of 0.4. We use PyTorch [3] to implement our model.

5.3 Open-world tail entity prediction

We compare our model MIA with other open-world KGC models, the experimental results are shown in Table 3. For a fair comparison, all the results are evaluated using target filtering.

The results for Target Filtering Baseline, DKRL and ConMask were obtained by the implementation provided by [30]. The Target Filtering Baseline assigns randomly scores to all entities that pass the target filtering. DKRL uses a two-layer convolutional neural network (CNN) over the entity descriptions. ConMask uses relationship-dependent content masking and fully convolutional neural network (FCN) to extract word-level target entity embedding from entity descriptions and then combine some other text features (entity names) are compared with the candidate tail entities to resolve a ranked list of candidate tail entities. Besides, we test the effect of the model without the attention mechanism between the candidates tail entities, which is marked as "Baseline" in Table 3.

As can be seen from the Table 3, our MIA model significantly outperforms Conmask in HITS@K, MR, and MRR by a large margin. At the same time, we also find that the MIA model performed better on the DBPedia50k dataset than on the FB20k dataset, because the entity description in the DBPedia50k dataset is more abundant than the entity description

[3] https://pytorch.org

Table 3 Open-world Tail Entity prediction results on FB20k and DBPedia50k

Model	DBPedia50k				
	HITS@1	HITS@3	HITS@10	MR	MRR
Target Filtering Baseline	0.045	0.097	0.23	104	0.11*
DKRL (2-layer CNN)	-	-	0.40	70	0.23
ConMask	0.47	0.65	0.81	16	0.58*
Baseline	0.65	0.84	0.93	5	0.76
MIA model	0.69	0.89	0.94	5	0.79
	$FB15k_{open}$				
Target Filtering Baseline	0.14	0.29	0.37	112	0.16
DKRL (2-layer CNN)	-	-	-	-	-
ConMask	0.24	0.45	0.63	71	0.33
Baseline	0.36	0.52	0.69	45	0.47
MIA model	0.41	0.55	0.71	39	0.51
	FB20K				
Target Filtering Baseline	0.17	0.32	0.41	123	0.27
DKRL (2-layer CNN)	-	-	-	-	-
ConMask	0.38	0.49	0.63	54	0.46
Baseline	0.46	0.64	0.81	20	0.58
MIA model	0.50	0.69	0.83	19	0.63

Note that we used the same evaluation protocol with target filtering as in ConMask. The asterisk (*) indicates that the result differs from the one published, because the MRR is calculated differently

in the FB20k dataset, where DBpedia50k dataset has an average entity description length of 454 words, FB20k dataset of 147 words. Besides, we can see that the performance on $FB15k_{open}$ is weaker than other datasets since entities do not exist in the training set, which exacerbates the data sparsity problem of FB15k. In addition, Table 3 shows that the "Baseline" is slightly less effective than the full model, demonstrating the effectiveness of the attention mechanisms between candidate tail entities.

5.4 Closed-world entity prediction

Our model can also work on the closed-world KGC since the open-world KGC adds additional constraints to the closed-world KGC. The dataset used in this part, denoted as $FB15k_{closed}$, also removes the 47 entities and the triples that contain them. Unlike $FB15k_{open}$, all entities in the test set must exist in the training set for $FB15k_{closed}$. As is shown in Tables 4 and 5, we compare the results of our model on the closed-world KGC with several models including "TransR" [21], "Jointly" [40], "SimplE" [16], "TuckER " [1] and "RotatE" [34]. Latest methods, including SimplE, TuckER, and RetatE, use a a filtering method [4] when calculating HITS: each candidate tail entity filters out the other correct candidate tail entities when calculating its ranking. Such filtering method will result in HITS values higher than those without filtering. So, for such latest methods, we contrast HITS with the filtering method in Table 5. And for traditional methods, including TransR,

Table 4 Effect comparison of closed-world KGC without the filtering method

Model	$\text{FB15k}_{\text{closed}}$		
	HITS@1	HITS@3	HITS@10
TransR	0.40	0.71	0.77
Jointly	0.48	0.74	0.82
ConMask	0.35	0.69	0.74
Baseline	0.53	0.75	0.86
MIA model	0.58	0.81	0.88

Jointly and ConMask, we contrast HITS without the filtering method in Table 4. We find that our model outperforms most of these baseline methods given that we enhance the interactions between model inputs with attention mechanism. For HITS@1, our method is weaker than TuckER and RotatE, probably because they can represent some of the special relations in FB15K better such as symmetric/antisymmetric relations. In addition, the results of ConMask and our model are better on $\text{FB15k}_{\text{closed}}$ than on $\text{FB15k}_{\text{open}}$ as a whole. This is probably because entities in $\text{FB15k}_{\text{closed}}$ are more consistently distributed in the test set and training set than in $\text{FB15k}_{\text{open}}$.

5.5 Different score functions

In order to better calculate the score corresponding to the representation of the head entity and the representation of the tail entity, we try a variety of scoring functions to do evaluations in (19)–(21). The experimental results on DBPedia50k are shown in Table 6 where $Score_{multiply}$ represents multiplying vectors directly. We can see that $Score_{linear}$, calculating the score by multiplying two vectors with an intermediate matrix, has the best performance, given that it can better simulate the interaction between the two representations. In order to illustrate the performance of these four score functions further, we demonstrate how the epochs of the training process affect the loss in Figure 4. We can see that $Score_{multiply}$ has the highest final loss, probably because simple vector multiplication does not yield the hidden interaction between two representations. Besides, $Score_{bilinear}$ and $Score_{trilinear}$ converge slowly and have higher final loss than $Score_{linear}$ due to their complexity. In contrast, $Score_{linear}$ has the best performance, because it can obtain the hidden interaction between two representations through a transformation matrix without high complexity.

Table 5 Effect comparison of closed-world KGC with the filtering method

Model	$\text{FB15k}_{\text{closed}}$		
	HITS@1	HITS@3	HITS@10
SimplE	0.66	0.77	0.84
TuckER	0.74	0.83	0.89
RotatE	0.75	0.83	0.88
Baseline	0.64	0.81	0.88
MIA model	0.68	0.83	0.90

Table 6 Effect comparison of different score functions

Function	HITS@1	HITS@3	HITS@10
$Score_{multiply}$	0.641	0.832	0.926
$Score_{linear}$	0.655	0.838	0.934
$Score_{bilinear}$	0.650	0.836	0.932
$Score_{trilinear}$	0.648	0.835	0.930

5.6 Interaction between multiple candidates

To investigate how candidate tail entities interact with each other through the attention mechanism, we visualize the weight matrix between multiple candidates in a running example under different number of epochs. As shown in Figure 5, n cells in the i-th row represent the weights of n candidates when calculating the hidden representation of the i-th candidate where the darker color indicates higher weights. The merged attention weights over multiple candidates helps to aggregate the useful information from the current candidate into its hidden representation. Taking c1 as an example, the five colors in the first row of the three matrices in Figure 5 represent the weights of c1-c5 when calculating the hidden representation of c1. We found that as the number of epochs increases, the model gives higher weights to c1, c2 and c5, probably because they are thought to be more helpful in enhancing the hidden representation of c1.

It may be difficult to understand the weight matrices if we only focus on these candidate descriptions. But when we look at the description of the head entity, we can see that the model emphasizes the candidate with "music genre", especially "pop music", when aggregating the information of the candidate tail entities. This may benefit from the previous attention layer generating the representations of candidate tail entity descriptions which are aware of the head entity description. The comparison with "Baseline" in Table 3 also demonstrates the effectiveness of the attention between multiple candidate tail entities.

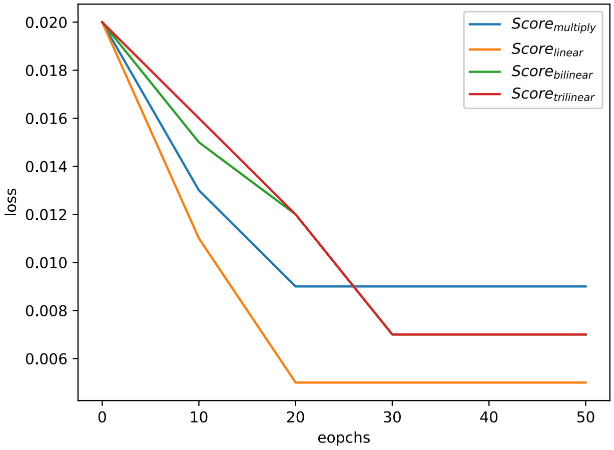
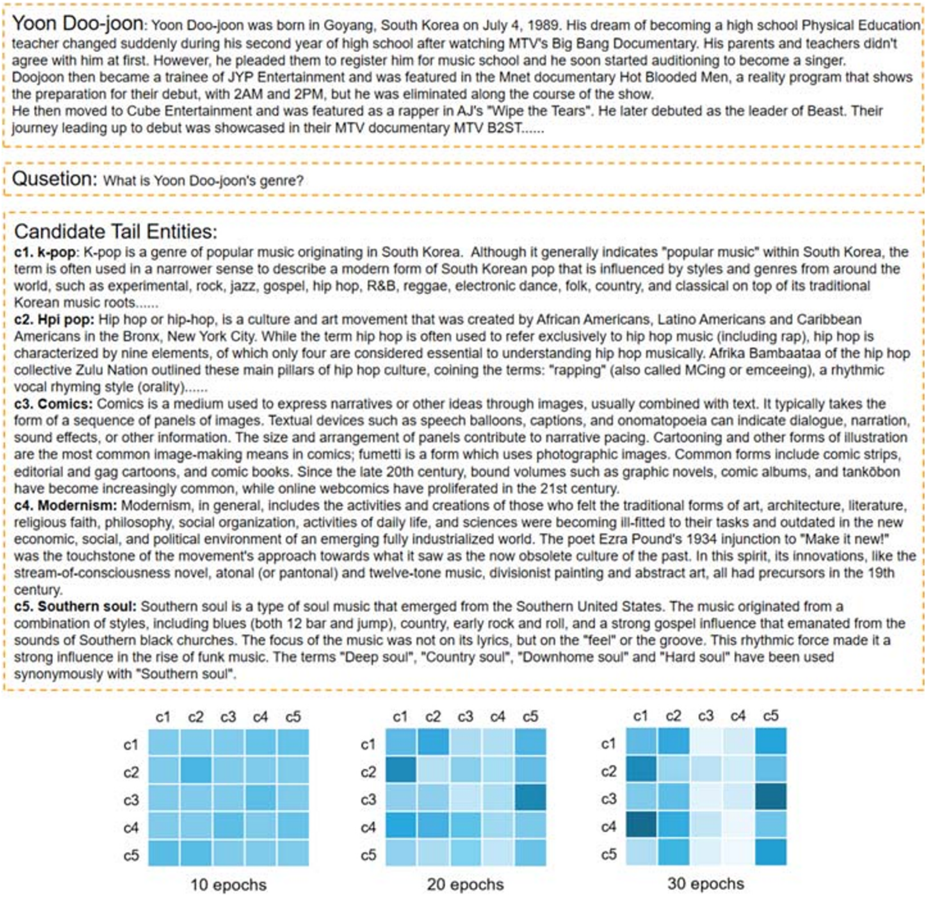


Figure 4 The loss of different score functions during training



Yoon Doo-joon: Yoon Doo-joon was born in Goyang, South Korea on July 4, 1989. His dream of becoming a high school Physical Education teacher changed suddenly during his second year of high school after watching MTV's Big Bang Documentary. His parents and teachers didn't agree with him at first. However, he pleaded them to register him for music school and he soon started auditioning to become a singer.
Doojoon then became a trainee of JYP Entertainment and was featured in the Mnet documentary Hot Blooded Men, a reality program that shows the preparation for their debut, with 2AM and 2PM, but he was eliminated along the course of the show.
He then moved to Cube Entertainment and was featured as a rapper in AJ's "Wipe the Tears". He later debuted as the leader of Beast. Their journey leading up to debut was showcased in their MTV documentary MTV B2ST......

Qusetion: What is Yoon Doo-joon's genre?

Candidate Tail Entities:
c1. k-pop: K-pop is a genre of popular music originating in South Korea. Although it generally indicates "popular music" within South Korea, the term is often used in a narrower sense to describe a modern form of South Korean pop that is influenced by styles and genres from around the world, such as experimental, rock, jazz, gospel, hip hop, R&B, reggae, electronic dance, folk, country, and classical on top of its traditional Korean music roots......
c2. Hpi pop: Hip hop or hip-hop, is a culture and art movement that was created by African Americans, Latino Americans and Caribbean Americans in the Bronx, New York City. While the term hip hop is often used to refer exclusively to hip hop music (including rap), hip hop is characterized by nine elements, of which only four are considered essential to understanding hip hop musically. Afrika Bambaataa of the hip hop collective Zulu Nation outlined these main pillars of hip hop culture, coining the terms: "rapping" (also called MCing or emceeing), a rhythmic vocal rhyming style (orality)......
c3. Comics: Comics is a medium used to express narratives or other ideas through images, usually combined with text. It typically takes the form of a sequence of panels of images. Textual devices such as speech balloons, captions, and onomatopoeia can indicate dialogue, narration, sound effects, or other information. The size and arrangement of panels contribute to narrative pacing. Cartooning and other forms of illustration are the most common image-making means in comics; fumetti is a form which uses photographic images. Common forms include comic strips, editorial and gag cartoons, and comic books. Since the late 20th century, bound volumes such as graphic novels, comic albums, and tankōbon have become increasingly common, while online webcomics have proliferated in the 21st century.
c4. Modernism: Modernism, in general, includes the activities and creations of those who felt the traditional forms of art, architecture, literature, religious faith, philosophy, social organization, activities of daily life, and sciences were becoming ill-fitted to their tasks and outdated in the new economic, social, and political environment of an emerging fully industrialized world. The poet Ezra Pound's 1934 injunction to "Make it new!" was the touchstone of the movement's approach towards what it saw as the now obsolete culture of the past. In this spirit, its innovations, like the stream-of-consciousness novel, atonal (or pantonal) and twelve-tone music, divisionist painting and abstract art, all had precursors in the 19th century.
c5. Southern soul: Southern soul is a type of soul music that emerged from the Southern United States. The music originated from a combination of styles, including blues (both 12 bar and jump), country, early rock and roll, and a strong gospel influence that emanated from the sounds of Southern black churches. The focus of the music was not on its lyrics, but on the "feel" or the groove. This rhythmic force made it a strong influence in the rise of funk music. The terms "Deep soul", "Country soul", "Downhome soul" and "Hard soul" have been used synonymously with "Southern soul".

Figure 5 Attention weight matrix visualization between multiple candidates

5.7 Ablation study

We carry out model ablations to further demonstrate the effectiveness of the proposed model. Firstly, we conduct an ablation analysis on the input word representation, which consists of several components: Part-Of-Speech (POS) Embedding, Named-Entity-Recognition (NER) Embedding and Handcrafted Features Embedding etc. The experimental results on DBPedia50k are shown in Table 7, we find all the input word representation components contribute to the performance of our MIA model. This suggests that it is useful to incorporate various feature into the word representation. We also remove our multiple interaction attention in the model. The results in Table 7 show a significant drop in performance by 1.5%, which indicates that the multiple interaction attention is effective in extracting the most relevant parts from the entity text description given different relationships.

Table 7 Ablations on several model components

Model	MRR
MIA model	0.770
w/o POS	0.766 (-0.004)
w/o Handcrafted Features	0.743 (-0.007)
w/o NER	0.762 (-0.008)
w/o Attention	0.755 (-0.015)

6 Conclusions and future work

This paper introduces an open-world KGC model called MIA that uses a word-level attention mechanism to simulate the interaction between the head entity description, head entity name, the relationship name and multiple candidate tail entity descriptions. In addition, we try to use additional textual features of head entity descriptions to enhance the head entity representation and apply the attention mechanism between candidate tail entities to enhance the representation of them. Besides, we try different scoring functions to increase the convergence of the model. Experiments on three datasets show that the MIA model has achieved significant improvement on the open-world KGC task compared to state-of-the-art models. However, MIA relies heavily on the richness of the entity descriptions, and the tail entity can be effectively predicted only when the necessary information related to the relationship is expressed in the entity description. In the future work, we consider to introduce more external knowledge into MIA to make it more robust.

Acknowledgments This research is partially supported by National Key R&D Program of China (No. 2018AAA0101900), the Priority Academic Program Development of Jiangsu Higher Education Institutions, National Natural Science Foundation of China (Grant No. 62072323, 61632016, 61836007, 61972069, 61832017, 61532018), Natural Science Foundation of Jiangsu Province (No. BK20191420), Natural Science Research Project of Jiangsu Higher Education Institution (No. 17KJA520003), and the Suda-Toycloud Data Intelligence Joint Laboratory.

References

1. Balažević, I., Allen, C., Hospedales, T.M.: Tucker: Tensor factorization for knowledge graph completion. arXiv preprint arXiv:1901.09590 (2019)
2. Bengio, Y., Simard, P., Frasconi, P., et al.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
3. 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, pp. 1247–1250. AcM (2008)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)
5. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading wikipedia to answer open-domain questions. arXiv preprint arXiv:1704.00051 (2017)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert:, Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Fu, C., Li, Z., Yang, Q., Chen, Z., Fang, J., Zhao, P., Xu, J.: Multiple interaction attention model for open-world knowledge graph completion. In: International Conference on Web Information Systems Engineering, pp. 630–644. Springer (2019)
8. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw. **18**(5-6), 602–610 (2005)

9. Gu, B., Li, Z., Zhang, X., Liu, A., Liu, G., Zheng, K., Zhao, L., Zhou, X.: The interaction between schema matching and record matching in data integration. IEEE Trans. Knowl. Data Eng. **29**(1), 186–199 (2016)
10. Hachey, B., Radford, W., Nothman, J., Honnibal, M., Curran, J.R.: Evaluating entity linking with wikipedia. In: AI, vol. 194, pp. 130–150. Elsevier (2013)
11. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
13. Huang, L., May, J., Pan, X., Ji, H., Ren, X., Han, J., Zhao, L., Hendler, J.A.: Liberal entity extraction: Rapid construction of fine-grained entity typing systems. Big Data **5**(1), 19–31 (2017)
14. 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 (Volume 1: Long Papers), vol. 1, pp. 687–696 (2015)
15. Kadlec, R., Schmid, M., Bajgar, O., Kleindienst, J.: Text understanding with the attention sum reader network. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2016)
16. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Advances in Neural Information Processing Systems, pp. 4284–4295 (2018)
17. Kingma, D.P., Ba, J.: Adam:, A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Lee, K., Salant, S., Kwiatkowski, T., Parikh, A., Das, D., Berant, J.: Learning recurrent span representations for extractive question answering. arXiv preprint arXiv:1611.01436 (2016)
19. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
20. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: EMNLP, pp. 705–714 (2015)
21. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI, vol. 15, pp. 2181–2187 (2015)
22. Liu, G., Wang, Y., Orgun, M.A.: Optimal social trust path selection in complex social networks. In: Twenty-Fourth AAAI Conference on Artificial Intelligence, pp. 1391–1398 (2010)
23. Liu, G., Wang, Y., Orgun, M.A., Lim, E.P.: Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks. IEEE Trans. Serv. Comput. **6**(2), 152–167 (2011)
24. Liu, G., Wang, Y., Orgun, M.A., Lim, E.P.: Finding the optimal social trust path for the selection of trustworthy service providers in complex social networks. IEEE Trans. Serv. Comput. **6**(2), 152–167 (2013)
25. Lukovnikov, D., Fischer, A., Lehmann, J., Auer, S.: Neural network-based question answering over knowledge graphs on word and character level. In: Proceedings of the 26th international conference on World Wide Web, pp. 1211–1220. International World Wide Web Conferences Steering Committee (2017)
26. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, vol. 2, pp. 1003–1011. Association for Computational Linguistics (2009)
27. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)
28. Qian, C., Zhu, X., Ling, Z.H., Si, W., Inkpen, D.: Enhanced lstm for natural language inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2017)
29. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
30. Shi, B., Weninger, T.: Open-world knowledge graph completion. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 1957–1964 (2018)
31. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, pp. 926–934 (2013)
32. Sordoni, A., Bachman, P., Bengio, Y.: Iterative alternating neural attention for machine reading
33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

34. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate:, Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
35. Trischler, A., Ye, Z., Yuan, X., He, J., Bachman, P., Suleman, K.: A parallel-hierarchical model for machine comprehension on sparse data
36. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M.: Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 417–426. ACM (2018)
37. Wang, L., Sun, M., Zhao, W., Shen, K., Liu, J.: Yuanfudao at semeval-2018 task 11:, three-way attention and relational knowledge for commonsense machine comprehension. arXiv preprint arXiv:1803.00191 (2018)
38. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
39. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
40. Xu, J., Chen, K., Qiu, X., Huang, X.: Knowledge graph representation with jointly structural and textual encoding. Arxiv Preprint Arxiv:1611.08661 (2016)
41. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
42. Zhang, D., Yuan, B., Wang, D., Liu, R.: Joint semantic relevance learning with text data and graph knowledge. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, pp. 32–40 (2015)
43. Zhang, Y., Liu, G., Liu, A., Zhang, Y., Li, Z., Zhang, X., Li, Q.: Personalized geographical influence modeling for poi recommendation. IEEE Intell Sys, (01), 1–1. https://doi.org/10.1109/MIS.2020.2998040 (2020)
44. Zhu, H., Wei, F., Qin, B., Liu, T.: Hierarchical attention flow for multiple-choice reading comprehension. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Lei Niu[1] · Chenpeng Fu[1] · Qiang Yang[2] · Zhixu Li[1,3] · Zhigang Chen[4] · Qingsheng Liu[5] · Kai Zheng[6,7]

Lei Niu
lniu@stu.suda.edu.cn

Chenpeng Fu
cpfu@stu.suda.edu.cn

Qiang Yang
qiang.yang@kaust.edu.sa

Zhixu Li
zhixuli@suda.edu.cn

Zhigang Chen
zgchen@iflytek.com

Qingsheng Liu
qsliu@iflytek.com

[1] Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou, China

[2] King Abdullah University of Science and Technology, Jeddah, Saudi Arabia

[3] iFLYTEK Research, Suzhou, China

[4] State Key Laboratory of Cognitive Intelligence, iFLYTEK, Suzhou, China

[5] Anhui Toycloud Technology, Hefei, China

[6] School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu, China

[7] School of Computer Science and Technology, Shandong University of Technology, Shandong, China