Triple Confidence-aware Encoder-Decoder Model for Commonsense Knowledge Graph Completion

Fu Zhang [a,*], Hongzhi Chen [a], Xiang Li [a] and Jingwei Cheng [a]
[a] *School of Computer Science and Engineering, Northeastern University, Shenyang, China*
E-mails: zhangfu@mail.neu.edu.cn, 2110693@stu.neu.edu.cn, 2101751@stu.neu.edu.cn, chengjingwei@mail.neu.edu.cn

Abstract.
Commonsense knowledge graphs have recently gained attention since they contain lots of commonsense triples, like (*get onto web, HasPrerequisite, turn computer on*), which usually use free-form text to represent the entities and are essential for many artificial intelligence applications. However, a large amount of valuable commonsense knowledge still exists implicitly or misses. In this case, commonsense knowledge graph completion (CKGC) is proposed to solve this incomplete problems by inferring the missing parts of the commonsense triples, e.g., (*?, HasPrerequisite, turn computer on*) or (*get onto web, HasPrerequisite, ?*). Some existing methods attempt to learn as much entity semantic information as possible by exploiting the structural and semantic context of entities for improving the performance of CKGC. However, we found that the existing models only pay attention to the entity and relation of the commonsense triple and ignore the important *confidence (weight)* information related to the commonsense triple. In this paper we innovatively introduce commonsense triple confidence into CKGC and propose a confidence-aware encoder-decoder CKGC model. In the *encoding* stage, we propose a method to incorporate the commonsense triple confidence into RGCN (relational graph convolutional network), so that the encoder can learn more accurate entity semantic representation by considering the triple confidence constraints. Moreover, as well known the commonsense knowledge graphs are usually sparse, because there are a large number of entities with an in-degree of 1 in the commonsense triples. Therefore, we propose to add a new relation (called similar edge) between two similar entities for compensating the sparsity of commonsense KGs. In the *decoding* stage, considering that the entities in the commonsense triples are sentence-level entities, we propose a joint decoding model by combining the InteractE and ConvTransE. Experiments show that our new model achieves better performance compared to the previous competitive models. In particular, the incorporating of the confidence scores of triples actually brings significant improvements to CKGC.

Keywords: Commonsense Knowledge Graph Completion, Triple Confidence, Encoder-Decoder Framework

1. Introduction

Since Google Knowledge Graph [1] was proposed in 2012, knowledge graphs (KGs), a.k.a. knowledge bases, have aroused considerable research interest. The structured knowledge called *facts* in KGs is organized in subject-predicate-object triples, also referred to as relations between head and tail entities. *Commonsense knowledge* is information that humans typically have that helps them make sense of everyday situations. As such, this knowledge,

*Corresponding author. Fu Zhang, PhD, Associate Professor, School of Computer Science and Engineering, Northeastern University, Shenyang, 110169, China, E-mail: zhangfu@mail.neu.edu.cn.

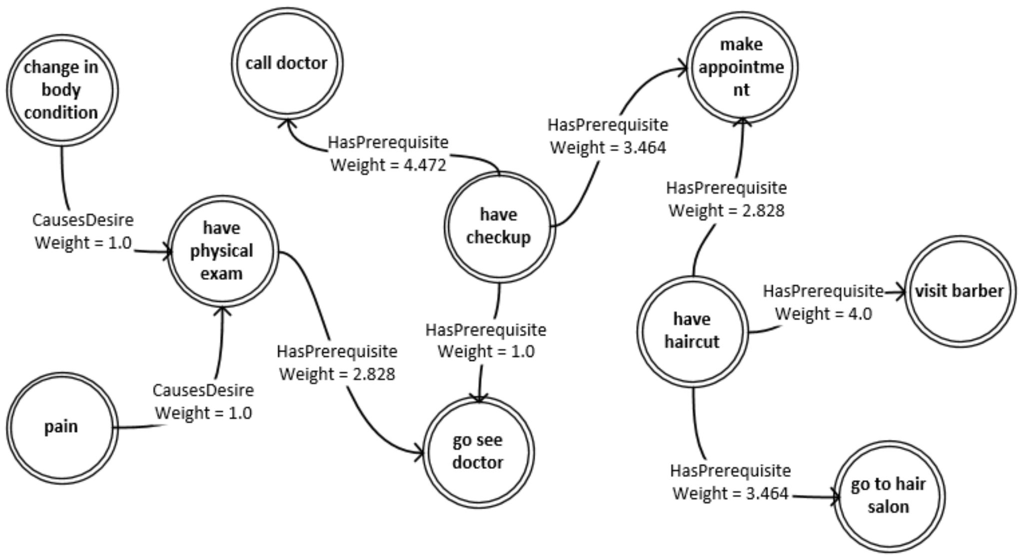

Fig. 1. Select part of the data in the popular commonsense knowledge graph ConceptNet-100K [10] to construct the subgraph. The *circle* represents the *node*, and the *directed edge* is composed of the *relation* and the *confidence* (*weight*) of a commonsense knowledge triple.

which can generally be assumed to be possessed by most people, is typically omitted in (written or oral) communication. The fact that commonsense knowledge is often implicit presents a challenge for automated natural language processing (NLP) [2] and question answering (QA) [3] approaches as the extraction and learning algorithms cannot count on the commonsense knowledge being available directly in text [4].

In order to complete the missing commonsense knowledge of facts, the *commonsense knowledge graph completion* (CKGC) is proposed to solve this incomplete problems, which is similar to the classical knowledge graph completion (KGC). Most of KGC methods adopt KG embedding techniques to predict the missing parts of the facts [5]. However, we found that at least the following challenges should be investigated and enhanced for CKGC:

1. As we have known, commonsense KGs are usually *sparse*, because there are a large number of entities with an in-degree of 1 in the triples of the commonsense KGs (as shown and analysed in Fig. 2 of Section 4). This can pose a challenge to typical KGC methods that learn entity/relation embedding representations solely from the knowledge that already exists in the graph.

   For compensating the sparsity of commonsense KGs for CKGC, the embedding of the pre-trained language model [6] and the textual entity identifiers [7] are used to develop entity embeddings that are more robust to sparsity. The dense processing of the commonsense KGs can also optimize the sparsity of the graphs to a certain extent. Treating all commonsense knowledge triples indiscriminately will also lead to inaccuracies in the integrated information. In some extent, it affects the effect of entity embedding.

2. In addition to the sparseness, another important difference between the commonsense and traditional KGs is that a *triple* in the commonsense KG may have a *confidence* (*weight*) as shown in Fig. 1. In this case, these *confidence values* can identify the importance of neighbor nodes of the node in the triple, and thus may be very useful for inferring the missing parts of the facts in CKGC. However, such confidence values *have not been explored and utilized* in previous work.

   Note that, the existing RGCN [8]-based commonsense knowledge graph completion model does not distinguish the importance of neighbor nodes when aggregating the neighbor node information of a node. The work in [7, 9] attempt to improve the entity embedding representation for CKGC by aggregating neighbor information. But the confidence values of triples in commonsense KGs as shown in Fig. 1 have never been utilized for CKGC in these previous work.

Based on the above observations, in this paper we propose an encoder-decoder CKGC model by innovatively incorporating commonsense triple confidence:

- We make a very detailed and deep experimental analysis regarding to the sparsity and confidence in the commonsense KG datasets (especially ConceptNet-100K [10]), some new observations are discovered, which can further make new inspiration for CKGC. The details can be found in Sections 4 and 6.
- We propose an encoder-decoder CKGC model by innovatively incorporating commonsense triple confidence. In the *encoding* stage, we propose a method to incorporate the commonsense triple confidence into RGCN. The encoder can aggregate the neighbor node information of the node in a triple, and more importantly it can distinguish the importance of neighbor nodes for well inferring the missing parts of the triple. Moreover, when any two entities in the commonsense KG are semantically similar, we propose to add a new relation (called similar edge) between two entities for compensating the sparsity of commonsense KGs.
- In the *decoding* stage, considering that the entities in the commonsense KG are sentence-level entities, we propose a joint decoding model by combining the InteractE [11] and ConvTransE [12].

Experiments show that our new model achieves better performance compared to the previous competitive models. In particular, the incorporating of the confidence values of triples actually brings significant improvements to CKGC.

2. Related Work

2.1. Knowledge Graph Completion (KGC)

Most of the existing knowledge graphs have incomplete problems, which can be alleviated by inferring missing links based on known facts. According to the triple structure of the knowledge graph (head entity, relation, tail entity), the main task of knowledge graph completion (KGC) is *entity prediction* (also called link prediction), which aims to predict the missing head entity or tail entity in the triple. In brief, the KGC methods can be roughly divided into *distance* model [13, 14], *hyperbolic space* model [15, 16], *tensor decomposition* model [17, 18] and *neural network* model [12, 19, 20]. The distance model defines distance-based scoring functions to compute the distance between two entities through the transformation of relation (e.g., TransE [13]). The hyperbolic space model embeds multi-relational graph data in the hyperbolic space, which can be thought of as a continuous analogue of discrete trees, making it suitable for modelling hierarchical data (e.g., MuRP [15]). The tensor decomposition model represents relations as linear transformations acting on entity vectors (e.g., DistMult [17], ComplEx [18]). The neural network model utilizes the popular neural networks as KG embedding techniques to predict the missing parts of the facts (e.g., SACN taking the benefit of GCN and ConvE together [12], ConvE [19]). Please refer to the survey [5] for more details and comparisons of these embedding methods.

2.2. Commonsense Knowledge Graph Completion (CKGC)

In 2020, Malaviya et al. [7] propose a model for complementing the commonsense knowledge graph. This is the first time that a specific model has been proposed for the completion of the commonsense knowledge graph, instead of directly using the traditional knowledge graph completion methods [10, 21, 22] to complete the commonsense knowledge graph. Malaviya et al. point out that the key challenge in completing commonsense KG is the scale and sparsity of the graph. For the problem of graph scale, Malaviya et al. use subgraphs for training to improve efficiency, thereby using the structure of the graph to provide complementary information to improve the completion performance. In response to the problem of sparsity, Malaviya et al. propose an approach for automatic graph densification based on semantic similarity scores between nodes. In addition to solving the above problems, Malaviya et al. adopt transfer learning from language models to commonsense knowledge graphs to improve contextual representation of nodes.

In 2021, InductivE [9] mainly point out that there are some entities in the test set and validation set that have not appeared in the training set, and thus propose the first benchmark for inductive commonsense KG completion task. Aiming at the problem of induction, InductivE leverage entity attributes based on transfer learning from word embedding, and the graph structure information aggregation through the relational graph convolutional neural network.

Inspired by the above methods, in this paper we innovatively propose to utilize the confidence of commonsense triples to help learn more accurate entity semantic representation by considering the triple confidence constraints. In the addition of similar edges in the graph structure, we decide whether to add or not by limiting the length of the path between two nodes. In the decoding stage, we use the joint convolution method to decode the obtained entity embedding and relation embedding, and then use the score function to predict the entity.

3. Problem Description

A commonsense knowledge graph is represent by $G = (N, V, C)$, where N is the set of nodes, V is the set of edges and C is the confidence of triple. It contains a set of *head entity, relation, tail entity* triples (h, r, t), where h is the head entity, t is the tail entity and r is the relation. The entity is defined in the graph as $E(G) = h|(h, r, t) \in G \cup t|(h, r, t) \in G$. Furthermore, $H(G, r) = h|(h, r, t) \in G$ and $T(G, r) = t|(h, r, t) \in G$ represent the head entity and tail entity of a relation.

The commonsense knowledge graph completion (CKGC) aims to answer accurately the queries with a timestamp $(?, r, t)$ or $(h, r, ?)$ by scoring higher for the true entity. After an incomplete triple is given, the model is used to find the correct entity among the limited candidate entities to complete the triple. An effective CKGC model should allow a large score difference between the correct entity and the wrong entity.

4. Datasets and Our Experimental Analysis

ConceptNet [23] is a knowledge graph which contains commonsense knowledge about the world, such as fact (*get onto web, HasPrerequisite, turn computer on*). In this paper we also utilize the popular commonsense knowledge graph completion (CKGC) dataset, i.e., ConceptNet-100k [10], which consists of the Open Mind Common Sense entries in the ConceptNet dataset. The abbreviation of ConceptNet-100K is CN-100K, where 100K represents the number of samples. For a fair comparation, in our work we utilize the training set, validation set and test set of ConceptNet-100k in the previous CKGC method [7].

4.1. Sparsity Analysis of ConceptNet-100K

As mentioned in Section 1, commonsense KGs are usually *sparse*. We use the node degree to explore the quantitative difference between the ConceptNet-100K dataset and the traditional KG dataset in entity prediction. Node degree is a measure of the edges (relations) linked to nodes (entities) in graph theory. Fig. 2 shows the cumulative frequency of 1 to 9 in-degrees for each dataset. It can be seen from the figure that the percentage of ConceptNet100K entities with degree 1 (84.00%) far exceeds that of FB15K [24]-237 (6.02%), WN18 [25] (14.87%) and FB15K [24] (2.18%). The degrees $\leqslant 2$ in ConceptNet100K is 91.43%, and the percentage in WN18 is just over half of 91.43%.

4.2. ConceptNet-100K-Confidence

Commonsense knowledge is a fact accepted by most people, but there are obvious differences in the reliability of the facts, such as (*get onto web, HasPrerequisite, turn computer on, 6.32*) and (*get onto web, HasPrerequisite, start your web browser, 1.0*) [1], where 6.32 is the confidence score of (*get onto web, HasPrerequisite, turn computer on* and 1.0 is the confidence score of (*get onto web, HasPrerequisite, start your web browser*). From the perspective of experimental comparison and fairness, we also continues to use the previous ConceptNet-100K dataset mentioned in [7], which is constructed based on the ConceptNet [23]. Note that, since the new ConceptNet (v5.7) knowledge graph [26] has more reliable confidence scores, we extract the confidence score of each triple in ConceptNet-100K

[1] The confidence level can be retrieved from the API provided by https://www.conceptnet.io/

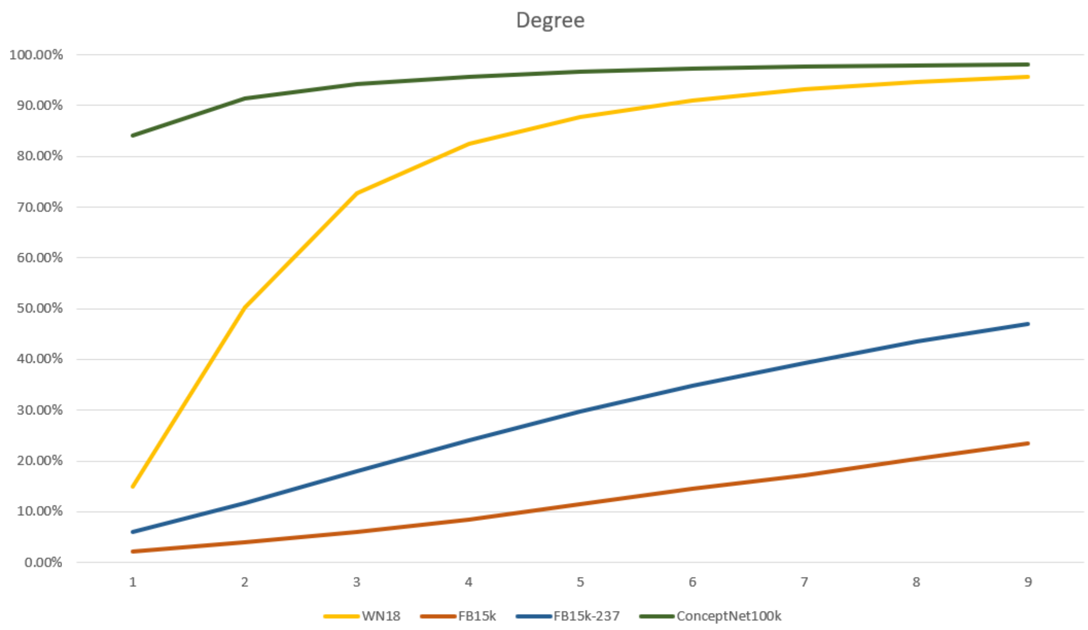

Fig. 2. Cumulative frequency of entity degree 1 to 9 in different datasets.

Table 1
In the ConceptNet-100K dataset, the confidence of the triples in train, validation and test

Matching method	Train(100000)	Validation(1200)	Test(1200)
Exact match	22235	709	782
Similarity match	24181	250	214
Failed match	53584	241	204

from it. But by comparing the two versions of ConceptNet, we found that the triples in previous CoceptNet [23] cannot be completely matched in the new ConceptNet (v5.7) [26]. In particularly, some triples in the previous ConceptNet have been adjusted, and even some relations no longer exist in ConceptNet (v5.7) knowledge graph, such as "NotIsA", "NotHasA" and "NotMadeOf". Table 1 shows an overview of how many triples in the dataset ConcepetNet-100K [10] can be matched.

Therefore, given each triple in ConceptNet-100K, we first find the same triple from the ConceptNet (v5.7) knowledge graph, and then get the confidence score of the triple. Then, the other unmatched triples are finally matched through the method of text similarity (by counting the number of identical words in two triples) and semantic similarity (by calculating the cosine similarity of two triples). Table 2 shows some examples of obtaining triple confidence through text similarity when a complete match is not possible. Table 3 shows some examples of obtaining triple confidence with a higher threshold through semantic similarity.

5. Our Model

The overall architecture of our model is shown in Fig. 3, which composed of encoder and decoder. In the *encoding* stage, we propose to incorporate the commonsense triple confidence into RGCN. The advantage is to avoid blindly aggregating the information of neighbor nodes. Our confidence-aware encoder can further improve the reliability of embedding. Before that, the pre-training language model is fine-tuned through the triples in the commonsense KG, so as to initialize the embedding of the entities in the commonsense KG. After getting the high-quality embedding,

Table 2

Commonsense confidence data obtained by text similarity matching in ConceptNet-100K

ConceptNet-100K			ConceptNet (v5.7)			
Head entity	Realtion	Tail entity	Head entity	Realtion	Tail entity	Weight
get onto web	HasPrerequisite	turn on your computer	get onto web	HasPrerequisite	turn computer on	1.000
most rock	HasProperty	hard	most rocks	HasProperty	hard	6.000
necklace	ReceivesAction	wear around neck	necklace	ReceivesAction	worn around neck	1.000
go to pub	UsedFor	have drink	going to pub	UsedFor	having drink	6.000
maintain good health	HasSubevent	exercise	maintain good health	HasSubevent	excersise	1.000

Table 3

Commonsense confidence data obtained through semantic similarity matching in ConceptNet-100K

ConceptNet-100K			ConceptNet (v5.7)			
Head entity	Realtion	Tail entity	Head entity	Realtion	Tail entity	Weight
buy food	MotivatedByGoal	you be hungry	buy food	MotivatedByGoal	hungry	3.464
work	NotHasProperty	fun	work does not involve thinking	NotHasProperty	interesting	1.000
diminish your own hunger	HasSubevent	you eat some food	diminish own hunger	HasSubevent	eat food	5.657
smoke	HasProperty	bad for you	smoke	HasProperty	harmful to animal's health	1.000
kill someone	Causes	go to jail	killing people	Causes	being sent to prison	1.000

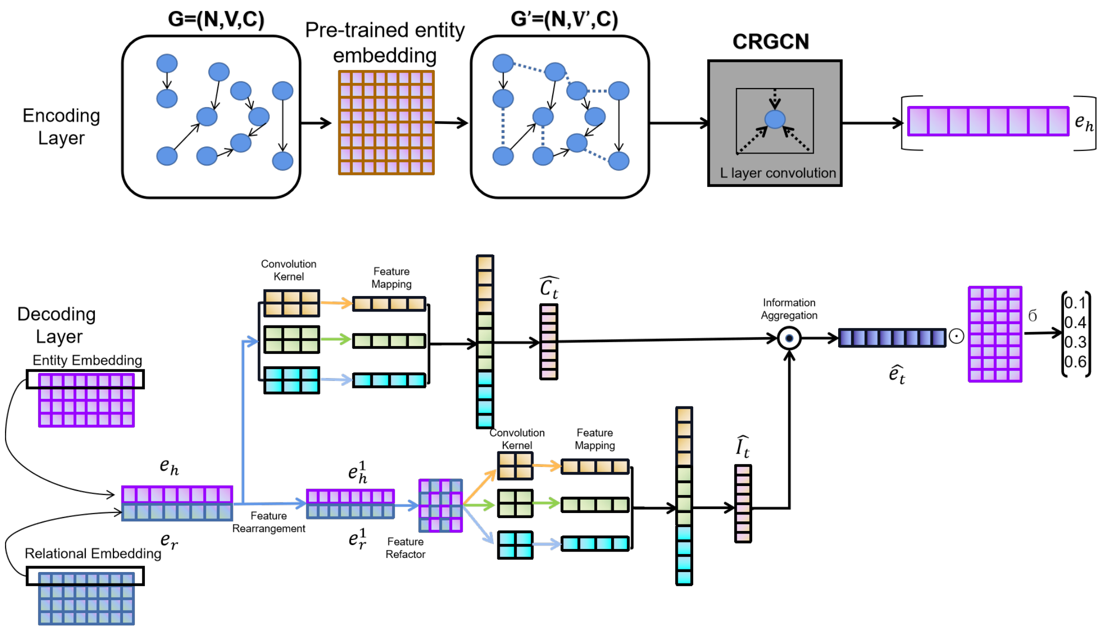

Fig. 3. Our confidence-aware encoder-decoder model for commonsense knowledge graph completion (CKGC). Here, CRGCN is our confidence relational graph convolutional networks by incorporating the commonsense triple confidence into RGCN. G represents the original graph constructed from commonsense triples. G' represents the graph after adding similar edges (dashed lines in G') for compensating the sparsity of commonsense KGs. $\hat{C}_t$ and $\hat{I}_t$ represent the vectors with the same embedding dimension of the entity obtained through ConvTransE and InteractE respectively. Relational embedding is obtained by random embedding.

in the *decoding* stage, considering that the entities in the commonsense KG are sentence-level entities, we propose a joint decoding model.

5.1. Encoding Structure

The encoding stage is mainly divided into the following several modules. First of all, the pre-training language model BERT [6] is used to obtain the *initial entity embeddings*. Then, we further add some similar edges to *increase the density of the graph* for relieving the sparsity of commonsense KGs. Final, the commonsense triple confidence is further introduced into RGCN for accurately aggregating the information of neighbor entities.

5.1.1. Initial Entity Embedding

In order to improve the quality of entity embedding, the selection of pre-training language models is also particularly important. The common ELMO [27] model is limited by the LSTM [28] neural network and cannot perform deep modeling work. The GPT model is limited by the partial structure of the decoder using Transformer [29], which leads to only attention The text information before the current word; The advantages of the BERT [6] model is that it is also a kind of representation learning, which can learn a high-quality text embedding through its own deep model structure. The pre-trained language model BERT has been proven to be a model that can improve the performance of natural language processing tasks through a deep network structure, and the model can also be obtained through unlabeled data training. For this reason, this paper utilizes an embedding method for entities in the commonsense knowledge graph based on the pre-trained language model BERT.

Inspired by [7] and [9], we also use fastText and fine-tune the BERT model with the masked language modeling task upon the set of textual entity identifiers for the commonsense knowledge graph. We apply the BERT-base uncased model to the commonsense entity identifiers and mean pooling across the token representations from all layers to obtain a feature embedding. The relationships in many triples are composed of multiple words, and the word segmentation database of the BERT model can actively perform word segmentation processing on the relationship. Initialized embedding continuous fine-tuning in the encoding stage, so that the correct candidate entity in the decoding stage can get a higher score.

5.1.2. Graph Density

In addition to the existing relationships between entities, some entities may have some similarities in semantics. For example, entities *"go see doctor"* and *"call doctor"* in Figure 1 are conceptually similar. The above example is a perfect display of similar relationships. There are actually many entities that are very similar but already have a certain relationship, such as *(pet, IsA, animal)* and *(pet, RelatedTo, animal)*. There are many ways to calculate the similarity between entities and introduce similar edges. The method proposed by Malaviya [7] is to perform global thresholding based on the similarity measure of the original node attributes. The above methods have high requirements for the quality of the initial embedding, and the number of similar edges cannot be controlled.

In our work, the addition of similar edges does not rely on the original initialization vector, and the calculation of similar edges is performed after multiple training. The entity embedding representation obtained by the aggregation of the weight relationship is of higher quality. In order to avoid the addition of false similar edges, we have made other restrictions in addition to the threshold. Before adding similar edges, we set the step distance between two entities not to exceed four. If the distance is too large, the similarity relationship is not considered. After obtaining the triples with similar edges, the weight of the triples is set to the size of the threshold. The ultimate goal of multiple restrictions is to increase the density of the graph while reducing the error caused by the wrong edge connection.

5.1.3. Confidence Relational Graph Convolutional Networks

Graph neural network can be understood as a simple differentiable message-passing framework [30] by Eq (1), where $h_i^{(l)}$ is the hidden state of node v_i in the *l-th* layer of the neural network, and $\mathcal{M}$ represents the incoming message set of node v_i, which is usually selected to be the same as the incoming edge set. $g_m(\cdot,\cdot)$ is typically chosen to be neural network-like function, and $\sigma(\cdot)$ is a element-wise activation function. The encoding model in this paper is mainly motivated as an extension of relational graph convolutional network [8], which was mainly proposed for large-scale relational data. The previous InductivE [9] model adaptively performs the aggregation operation of neighbor nodes by adding gating units. The gating function mainly controls the flow of information based on the interaction between the center and neighboring nodes. The adaptive method improves the performance of the model to a certain extent, but it ignored the authenticity and reliability of commonsense knowledge.

$$h_i^{l+1} = \sigma(\sum_{m \in \mathcal{M}} g_m(h_i^{(l)}, h_j^{(l)})) \tag{1}$$

This paper uses a relational graph convolutional neural network that quotes the confidence of triples. The confidence relational graph convolutional encoder takes graph G as input and encodes each node as a D-dimensional embedding $h_i \in \mathcal{R}^D$ for all nodes $h, r \in N$. Given a graph G with R relationship types and a GCN with L layer[2], the operation to calculate the entity embeddding of the node h, r in the $l+1$ layer is:

$$h_i^{l+1} = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} C_{i,j} W^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)}) \tag{2}$$

where $h_j^l \in \mathcal{R}^{d^{(l)}}$ and $h_i^l \in \mathcal{R}^{d^{(l)}}$ are the hidden states of the neighbor nodes of node n_i and node n_i in layer l-th of confidence relational graph convolutional networks respectively, with $d^{(l)}$ being the dimension of the l-th layer. $C_{i,j}$ represents the confidence value of node n_i and node $node_j$ under the relationship r, and W^l is a linear projection matrix specific to the lth layer. The node n_i information and the neighbor nodes information of node n_i are accumulated and passed through the element-wise activation function $\sigma(\cdot)$.

5.2. Decoding Structure

The encoding stage mainly optimizes the entity embedding, and then obtains a low-dimensional entity embedding representation that combines entity semantic information and structural information, and then selects different models and score functions in the decoding stage to perform negative feedback adjustment. As mentioned above, the nodes (entities) of the commonsense knowledge graph are composed of free text. The node vector expressed by embedding can be understood as a sentence vector, and thus the expressive ability of the neural network needs to be considered. Additionally, the data scale of the commonsense knowledge graph also requires the parameters of the neural network model to be optimized. From the perspective of expressive ability and model processing efficiency, we propose a multi-level convolutional neural network model in the decoding stage.

5.2.1. Single Convolutional model

The ConvTransE [12] model retains the translation invariance of TransE [13] ($e_h + e_r \approx e_t$[3]) in structure. The translation invariance is mainly reflected in the operation of ConvTransE, which can be understood as the accumulation of e_h and e_r after one-dimensional convolution (Eq (3)) . In the embedding, the dimensions of the entities embedding and relations embedding are the same, and the input of ConvTransE is a stack of entity embedding and relation embedding, where $c$represents the c-th convolution kernel, and w_c is the parameter of the c-th convolution kernel. K represents the volume The width of the product core, and n is the index of the output vector ($n \in [0, d-1]$), where d is the dimension of the embedding representation. $\hat{e_h}$ represents the head entity embedding after padding, $\hat{e_r}$ represents The tail entity is embedded after padding. The final output vector is $M_c(e_h, e_r) = [m_c(e_h, e_r, 0), ..., m_c(e_h, e_r, d-1)]$.

$$m_c(e_h, e_r, n) = \sum_{\tau=0}^{K-1} w_c(\tau, 0)\hat{e_h}(n+\tau) + w_c(\tau, 0)\hat{e_r}(n+\tau) \tag{3}$$

The original intention of the InteractE[11] model is to enhance the expression ability of the ConvE [19] model. The commonsense knowledge graph entity embedding contains rich semantic information and can be extracted with the help of InteractE. The model mainly uses three methods to enhance the information interaction between

---

[2]The number of convolutional layers in this paper is 2

[3]e_h, e_r, e_trespectively represent the embedding of the head entity, the embedding of the relationship and the embedding of the entity.

embedded representations. First, it use random arrangement to reorder the input embedded representations instead of the fixed ordering method. If you use t different ordering methods, you can get t different interactive information $P_t = [(e_h^1, e_r^1), ..., (e_h^t, e_r^t)]$. Second, it embed the reordered entities and relationships to represent the improvement of heterogeneous interaction through shape reshaping function:

$$\emptyset(P_t) = [\emptyset(e_h^1, e_r^1), ..., \emptyset(e_h^t, e_r^t)] \tag{4}$$

Third, it use the convolution operation of circular convolution (Eq (5)) to replace the standard convolution operation, where I represents the input after reshaping the shape, and W represents a convolution kernel of size k×k. When using the padding operation to fill the input content, the ordinary filling method directly fills the input with 0, and the circular convolution fills in the upper, lower, left, and right contents.

$$[I * W]_{p,q} = \sum_{i=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{j=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} I_{[p-i]_m, [q-i]_n} W_{i,j} \tag{5}$$

5.2.2. *Joint Convolutional Model*

This paper takes into account that the entity representation in the commonsense knowledge graph is free text, and combines the characteristics of ConvTransE [12] and InteractE [11] to propose a joint convolution model (down side of Fig. 2). The following formula maps the two vectors with the same dimension as the tail entity embedding through MLP:

$$\psi(e_h, e_r) = MLP(C_t, I_t) \tag{6}$$

where C_t (Eq (7)) is the embedding vector obtained by ConvTransE, and I_t (Eq (8)) is the embedding vector obtained by InteractE. The $\star$ indicates the deep convolution operation using circular convolution.

$$C_t = f(vec(M(e_h, e_r))W) \tag{7}$$

$$I_t = g(vec(g(\emptyset(P_t) \star w))W) \tag{8}$$

In the joint convolution model, more interactive InteractE [11] convolution operations and ConvTransE [12] convolution operations are performed respectively in the embedding of the input head entity and relationship. In the convolution operation using InteractE, the first step is to perform feature sorting operations. In order to capture various heterogeneous interactions, t random permutations of e_h and e_r are interactively generated by $P_t = [(e_h^1, e_r^1), ..., (e_h^t, e_r^t)]$ to indicate. Under high probability, the set of interactions in $\emptyset(e_h^i, e_r^i)$ will not be repeated, because the embedding dimension is very large, so the number of different interactions in all possible arrangements is very large . Therefore, for t different arrangements, the total number of interactions can be expected to be approximately t times the number of interactions. The second step is to adjust the arrangement. Arrange the embedding through rectangles of equal height and equal width, and ensure that the embedding of the entity and the embedding of the relationship are not adjacent. In this way, the largest heterogeneous interaction between entity and relationship features is captured. The third step is the circular convolution operation. Compared with the traditional convolution operation, the circular convolution can further strengthen the interaction. Interaction stacks each reshaped arrangement into a separate channel. For convolution arrangement, circular convolution is applied in a depth calculation method. Although different filters can be applied to each permutation, it is found that sharing filters across channels is better in actual practice, because it allows a set of kernel weights to be trained on more input instances. In the convolution operation using ConvTransE, the head entity and tail entity are composed of shorter text and one-to-one triples, which will be better than the depth convolution of InteractE rearrangement. Therefore, a joint convolution operation is performed combining the advantages of the two convolution operations.

5.3. Training

For a commonsense KG, we also consider the use of inverse relations to increase the scale of the commonsense KG. For example, the inverse relation triple of (h, r, t) is (t, r^{-}, h). Given an incomplete triple $(h, r, ?)$ (or $(?, r, t)$), the model is used to calculate the score of candidate entities in the commonsense knowledge graph. An efficient model should calculate that the score of the correct tail entity is much higher than the score of the wrong tail entity. In the ranking of the entity, not only the positive ranking is considered, for example, the tail entity is predicted by $(h, r, ?)$. We also consider the ranking situation obtained by the inverse relation, such as predicting the head entity by $(t, r^{-}, ?)$. Each result of the final ranking takes the average of the ranking of the head entity and the ranking of the tail entity.

Before training, we need to define the scoring function. In the completion task, an incomplete triple *(h, r, ?)* needs to be given. The *D*-dimensional embedding of the head entity is obtained through the encoding layer, and the embedding of the relationship is obtained by random embedding representation. After two single convolution models in the decoder, the code with the same dimension as the entity embedding can be obtained respectively. Two single convolution models in the decoder, the codes with the same dimensions as the entity embedding can be obtained respectively. After the two codes obtained are mapped through the fully connected layer, the final vector of the same dimension as the entity embedding is obtained, and the obtained embedding representation is set as $\hat{e}_t$, and the candidate tail entity is represented by the embedding matrix $E_{entity\_num*entity\_dimension}$. Finally, the obtained embedding representation $\hat{e}_t$ and the candidate tail entity matrix are multiplied by a product operation (Eq (9)) to obtain the score of each candidate tail entity. Using the above-mentioned matrix product method can efficiently calculate the scores of multiple candidate sets at the same time and improve the efficiency of the completion network.

$$Score = \hat{e}_t * E_{entity\_num*entity\_dimension} \tag{9}$$

This article uses a 1vsAll training strategy [31] with a binary cross-entropy loss function, which can be understood as a multi-classification problem. The effective use of graph convolutional neural networks needs to solve the problem of the large scale of the graph, so this paper uses the method of subgraph sampling to control the scale of the graph. To limit the number of neighbor nodes sampled in each layer, we use the method of sampling by layer. The core idea of hierarchical sampling is to limit the number of nodes sampled in each layer, so that the number of neighbor nodes will only increase linearly as the number of layers increases. Each training data in the dataset is in the format of *(h, r, t)*, and the input part in the model is *(h, r, ?)*. Calculate the score for each entity through the above-mentioned entity score calculation method, and use the activation function sigmoid to map the score value in the *[0,1]* interval. The entity outside the complete fact triple is regarded as the wrong candidate entity, and the binary cross-entropy loss function (Eq (10)) is used for calculation, where *N* represents the number of input triples, *input* is triples, and *label* is the correct label for each triple. In this paper, the Adam [32] optimization function combined with decoupled weight decay regularization [33] and label smoothing are used to adjust the parameters of the model. The model is trained for multiple iterations. If it is verified that the mean reciprocal rank (MRR) does not improve under 15 iterations, the training will be terminated early. A single NVIDIA GeForce GTX3090 was used to train all models used in this commonsense knowledge graph completion model.

$$Loss(input, label) = -\frac{1}{N}\sum_i (label_i \cdot log(input_i) + (1 - label_i) \cdot log(1 - input_i)) \tag{10}$$

6. Experiments and Results

6.1. Baselines

In order to make the comparison model convincing, this paper adopts a series of comparison models proposed by Malaviya et al. (2020) [7]: DistMult [17], ComplEx [18], ConvE [19] and ConvTransE [12]. The performance of

Table 4
Evaluation results on ConceptNet-100K. The upper part is the completion results of some benchmark models, [♣] represents the result from [7], [♠] represents the result from [9]. Best results are in bold.

ConceptNet-100K				
Model	MRR	Hits@1	Hits@3	Hits@10
Distmult [♣]	8.97	4.51	9.76	17.44
Complex [♣]	11.40	7.42	12.45	19.01
Conve [♣]	20.88	13.97	22.91	34.02
Convtranse [♣]	18.68	7.87	23.87	38.95
Comet-normalized [♣]	6.07	0.08	2.92	21.17
Interacte	22.70	14.60	15.30	35.20
Malaviya et al. [♣]	51.11	39.42	59.58	73.59
Inductive [♠]	57.35	-	**64.50**	78.00
Our model	55.43	44.05	62.30	76.50
Our model + Confidence	**57.45**	**46.13**	64.00	**79.13**

these models is not necessarily the best in the completion of the commonsense knowledge graph. The above models have absolute advantages after being screened to illustrate the effectiveness of the confidence level of commonsense knowledge proposed in this paper.

In addition to the aforementioned baseline model, we also introduced the latest InductivE [9] model and InteractE [11] model for comparison. But using the latest benchmark model to compare with the previous model, by adjusting the training parameters and changing the corresponding embedding, it can definitely exceed the original model. In order to avoid the above situation, we use our own model as the benchmark model, and introduce the model into commonsense confidence and not introduce commonsense confidence to conduct experiments to illustrate the importance of confidence.

6.2. Evaluation Metrics

When using the entity prediction task to complete the commonsense knowledge graph, it is usually evaluated by sorting out the score rankings of the triples. The specific evaluation indicators used are as follows. The first is *Hits@n*(n=1, 3, 10), which is the case where the correct triples are ranked in the top *n* among all combinations of triples. If the value of *Hits@1* is larger, the accuracy of the model is higher, and *Hits@3* and *Hits@10* can indicate the accuracy of the model. The second is the MR (Mean Rank) indicator. The results of the average processing of all the triples rankings, we can see how well the model fits all the triples in the completion. If the value of MR is small, it means that the fit of the model is better. If the MR is A large value indicates that the model has differentiated in the evaluation, resulting in low scores for some correct triples. The third is the *MRR*(Mean Reciprocal Rank) indicator, which calculates the mean value of the inverse of the ranking of all triples, which reflects the overall effect of model completion. At the same time, following the past processing method, the correct entity that is not the target entity is removed by filtering before calculating the ranking of the candidate entity.

6.3. Result

The results of the experiment are shown in Table 4. It can be seen that our new model achieves better performance compared to the previous competitive models in the evaluation indicators *MRR*, *Hits@1* and *Hits@10*. In particular, based on our ablation study, the results show that the incorporating of the triple confidence into our model actually brings significant improvements to CKGC.

Moreover, through the in-depth study of the test results, we found that the incompleteness of the training samples also has a great impact on the test results. We select five triples in the test set where the tail entity prediction is ranked tenth as shown in Table 5. For example, the triple to be predicted is *(jellyfish, AtLocation, most ocean)*, which ranks tenth in the candidate set, but the triples *(jellyfish, AtLocation, sea water)*, *(jellyfish, AtLocation, coral reef)*, *(jellyfish, AtLocation, open ocean)*, and *(jellyfish, AtLocation, saltwater)* exist in ConceptNet but not in ConceptNet-

Table 5

This table mainly explains the impact of incompleteness of ConceptNet-100K on the model. We selected five examples where the triples to be predicted ranked tenth in the candidate set, and the triples marked with ∗ indicate that they exist in ConceptNet but do not exist in ConceptNet-100K

ConceptNet-100K test set					
Triples to be predicted	**(jellyfish, AtLocation, most ocean)**	**(human, AtLocation, park)**	**(chicken, IsA, meat)**	**(cat, HasA, whisker)**	**(water, UsedFor, drink)**
Candidate_1	(jellyfish, AtLocation, sea)	(human, AtLocation, build)	(chicken, IsA, poultry) ∗	(cat, HasA, pet)	(water, UsedFor, swim)
Candidate_2	(jellyfish, AtLocation, indian ocean)	(human, AtLocation, town)	(chicken, IsA, pet)	(cat, HasA,leg)	(water, UsedFor, relation)
Candidate_3	(jellyfish, AtLocation, any ocean)	(human, AtLocation, new york)	(chicken, IsA, mammal)	(cat, HasA, nose) ∗	(water, UsedFor, wash your hand)
Candidate_4	(jellyfish, AtLocation, gulf of mexico)	(human, AtLocation, room) ∗	(chicken, IsA, vegetarian food)	(cat, HasA, feel)	(water, UsedFor, wash off)
Candidate_5	(jellyfish, AtLocation, sea water) ∗	(human, AtLocation, train station)	(chicken, IsA, vegetarian)	(cat, HasA, two ear)	(water, UsedFor, take bath) ∗
Candidate_6	(jellyfish, AtLocation, in ocean)	(human, AtLocation, paris)	(chicken, IsA, egg)	(cat, HasA, foot)	(water, UsedFor, grow vegetable)
Candidate_7	(jellyfish, AtLocation, coral reef) ∗	(human, AtLocation, theatre) ∗	(chicken, IsA, roast)	(cat, HasA, tooth)	(water, UsedFor, take bath in)
Candidate_8	(jellyfish, AtLocation, open ocean) ∗	(human, AtLocation, street)	(chicken, IsA, usually)	(cat, HasA, fur to protect it skin)	(water, UsedFor, eat)
Candidate_9	(jellyfish, AtLocation, saltwater) ∗	(human, AtLocation, factory) ∗	(chicken, IsA, food animal)	(cat, HasA, cat hair)	(water, UsedFor, cool off)
Candidate_10	**(jellyfish, AtLocation, most ocean)**	**(human, AtLocation, park)**	**(chicken, IsA, meat)**	**(cat, HasA, whisker)**	**(water, UsedFor, drink)**

Table 6

This table shows the comparison of the results of this model after introducing the filtering mechanism constructed by complete data

ConceptNet-100K					
Model	MR	MRR	Hits@1	Hits@3	Hits@10
Our model + Confidence	127.34	57.45	46.13	64.00	79.13
Our model + Confidence + Complete filtering	**125.88**	**58.31**	**47.22**	**64.52**	**79.45**

100K. The above phenomenon of incomplete data affects the filtering mechanism, and thus affects the model's effect on the completion of the commonsense knowledge graph. More results can be found in the Appendix.

In order to solve the impact of the above-mentioned data incompleteness on the completion effect of the commonsense knowledge graph, we carried out comparison experiments by loading the complete ConceptNet dataset. The main purpose of loading the complete CoceptNet dataset is to obtain complete commonsense triples data. They filter all the correct triples in the filtering mechanism, and they do not participate in training. Table 6 shows the comparison of the results of this model after introducing the filtering mechanism constructed by complete data. It can be seen from the table that all indicators have been improved. The introduction of MR evaluation indicators makes the improved results more intuitive. Combining Table 4 and Table 6, it can be found that after the introduction of complete data for filtering, the evaluation indicators have a certain degree of improvement over all benchmark models.

7. Conclusion

In this paper we propose a confidence-aware encoder-decoder model for commonsense knowledge graph completion (CKGC). Our work is the first work to introduce commonsense triple confidence into CKGC, in order that the model can integrate more recognizable neighbor entity information to learn more accurate entity semantic representation. Moreover, we also propose to add a new relation (called similar edge) between two similar entities for compensating the sparsity of commonsense knowledge graphs. In addition, considering that the entities in the commonsense triples are sentence-level entities, we propose a joint decoding model by combining the advantages of InteractE and ConvTransE. Experiments show that our new model achieves better performance compared to the previous competitive models. In particular, the incorporating of the confidence scores of triples actually brings significant improvements to CKGC.

References

[1] A. Singhal, Introducing the knowledge graph: things, not strings, *Official google blog* **5** (2012), 16.

[2] S. Vashishth, R. Joshi, S.S. Prayaga, C. Bhattacharyya and P. Talukdar, Reside: Improving distantly-supervised neural relation extraction using side information, *arXiv preprint arXiv:1812.04361* (2018).

[3] A. Bordes, S. Chopra and J. Weston, Question answering with subgraph embeddings, *arXiv preprint arXiv:1406.3676* (2014).
[4] F. Ilievski, A. Oltramari, K. Ma, B. Zhang and P. Szekely, Dimensions of Commonsense Knowledge, *arXiv preprint arXiv:2101.04640* (2021).
[5] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Transactions on Knowledge Data Engineering* **29**(12) (2017), 2724–2743.
[6] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
[7] C. Malaviya, C. Bhagavatula, A. Bosselut and Y. Choi, Commonsense knowledge base completion with structural and semantic context, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 2020, pp. 2925–2933.
[8] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. Van Den Berg, I. Titov and M. Welling, Modeling relational data with graph convolutional networks, in: *European semantic web conference*, Springer, 2018, pp. 593–607.
[9] B. Wang, G. Wang, J. Huang, J. You, J. Leskovec and C.-C.J. Kuo, Inductive learning on commonsense knowledge graph completion, in: *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, pp. 1–8.
[10] X. Li, A. Taheri, L. Tu and K. Gimpel, Commonsense knowledge base completion, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1445–1455.
[11] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal and P. Talukdar, Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 2020, pp. 3009–3016.
[12] C. Shang, Y. Tang, J. Huang, J. Bi, X. He and B. Zhou, End-to-end structure-aware convolutional networks for knowledge base completion, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 3060–3067.
[13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, Translating embeddings for modeling multi-relational data, *Advances in neural information processing systems* **26** (2013).
[14] G. Ji, S. He, L. Xu, K. Liu and J. Zhao, 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)*, 2015, pp. 687–696.
[15] I. Balazevic, C. Allen and T. Hospedales, Multi-relational poincaré graph embeddings, *Advances in Neural Information Processing Systems* **32** (2019), 4463–4473.
[16] I. Chami, A. Wolf, F. Sala and C. Ré, Low-dimensional knowledge graph embeddings via hyperbolic rotations, in: *Graph Representation Learning NeurIPS 2019 Workshop*, 2019.
[17] B. Yang, W.-t. Yih, X. He, J. Gao and L. Deng, Embedding entities and relations for learning and inference in knowledge bases, *arXiv preprint arXiv:1412.6575* (2014).
[18] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier and G. Bouchard, Complex embeddings for simple link prediction, in: *International conference on machine learning*, PMLR, 2016, pp. 2071–2080.
[19] T. Dettmers, P. Minervini, P. Stenetorp and S. Riedel, Convolutional 2d knowledge graph embeddings, in: *Thirty-second AAAI conference on artificial intelligence*, 2018.
[20] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen and D. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, *arXiv preprint arXiv:1712.02121* (2017).
[21] I. Saito, K. Nishida, H. Asano and J. Tomita, Commonsense knowledge base completion and generation, in: *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 2018, pp. 141–150.
[22] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz and Y. Choi, Comet: Commonsense transformers for automatic knowledge graph construction, *arXiv preprint arXiv:1906.05317* (2019).
[23] R. Speer and C. Havasi, ConceptNet 5: A large semantic network for relational knowledge, in: *The People's Web Meets NLP*, Springer, 2013, pp. 161–176.
[24] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.
[25] G.A. Miller, WordNet: a lexical database for English, *Communications of the ACM* **38**(11) (1995), 39–41.
[26] R. Speer, J. Chin and C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: *Thirty-first AAAI conference on artificial intelligence*, 2017.
[27] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, Deep contextualized word representations, *arXiv preprint arXiv:1802.05365* (2018).
[28] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural computation* **9**(8) (1997), 1735–1780.
[29] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, Improving language understanding with unsupervised learning (2018).
[30] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals and G.E. Dahl, Neural message passing for quantum chemistry, in: *International conference on machine learning*, PMLR, 2017, pp. 1263–1272.
[31] D. Ruffinelli, S. Broscheit and R. Gemulla, You can teach an old dog new tricks! on training knowledge graph embeddings, in: *International Conference on Learning Representations*, 2019.
[32] D.P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
[33] I. Loshchilov and F. Hutter, Decoupled weight decay regularization. arXiv, *Preprint published January* **4** (2019).
[34] L. Yao, C. Mao and Y. Luo, KG-BERT: BERT for knowledge graph completion, *arXiv preprint arXiv:1909.03193* (2019).
[35] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A.H. Miller and S. Riedel, Language models as knowledge bases?, *arXiv preprint arXiv:1909.01066* (2019).

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, in: *Advances in neural information processing systems*, 2017, pp. 5998–6008.

Appendix A. Appendix

Table 7: The table lists the triples where the tail entity to be predicted in the test set does not enter the top ten, and the situation of the first candidate entity in the prediction

Triples to be predicted			Model prediction result		
Head entity	Realtion	Tail entity	Head entity	Realtion	Tail entity
baseball	IsA	sport	baseball	IsA	consider all-american sport
baseball	IsA	game	baseball	IsA	consider all-american sport
dog	CapableOf	bark	dog	CapableOf	poop
dog	IsA	mammal	dog	IsA	common pet
dog	AtLocation	kennel	dog	AtLocation	zoo
dog	CapableOf	guide blind person	dog	CapableOf	poop
dog	CapableOf	guard house	dog	CapableOf	poop
dog	HasA	four leg	dog	HasA	eye
dog	IsA	loyal friend	dog	IsA	common pet
dog	IsA	canine	dog	IsA	common pet
dog	CapableOf	guard your house	dog	CapableOf	poop
dog	Desires	chew on bone	dog	Desires	walk
dog	NotCapableOf	sweat	dog	NotCapableOf	walk
polo	IsA	game	polo	IsA	british sport
book	ReceivesAction	write	book	ReceivesAction	store in library
book	AtLocation	your desk	book	AtLocation	bookstore
book	AtLocation	classroom	book	AtLocation	bookstore
book	HasA	story	book	HasA	lot of information
paper	HasProperty	recyclable	paper	HasProperty	transparent
sex	Causes	child	sex	Causes	procreation
metal	IsA	music	metal	IsA	common metal
sushi	IsA	food	sushi	IsA	raw seafood
food	AtLocation	table	food	AtLocation	dinner
telephone	AtLocation	your desk	telephone	AtLocation	house
telephone	UsedFor	communication	telephone	UsedFor	make call
sleep	HasPrerequisite	close eye	sleep	HasPrerequisite	rest
water	AtLocation	pool	water	AtLocation	freeze
water	CapableOf	reflect image	water	CapableOf	melt
flower	AtLocation	park	flower	AtLocation	store
child	CapableOf	share toy	child	CapableOf	play with toy
computer	UsedFor	work	computer	UsedFor	do math
computer	UsedFor	play game	computer	UsedFor	do math
computer	IsA	electronic device	computer	IsA	excellent source of information
computer	AtLocation	your house	computer	AtLocation	any large city
computer	CapableOf	process information	computer	CapableOf	do math
apple	AtLocation	apple tree	apple	AtLocation	store
clothe	AtLocation	drawer	clothe	AtLocation	hanger
glass	UsedFor	drink	glass	UsedFor	correct poor vision
library	UsedFor	do research	library	UsedFor	study
key	CapableOf	open lock	key	CapableOf	unlock door
key	CapableOf	open door	key	CapableOf	unlock door
key	UsedFor	open door	key	UsedFor	unlock old door
cat	CapableOf	hunt mouse	cat	CapableOf	bite
cat	CapableOf	drink water	cat	CapableOf	bite
cat	AtLocation	lap	cat	AtLocation	home
cat	HasA	whisker	cat	HasA	leg
cat	AtLocation	windowsill	cat	AtLocation	home
cat	CapableOf	catch mouse	cat	CapableOf	bite
cat	CapableOf	corner mouse	cat	CapableOf	bite
wood	UsedFor	fence in property	wood	UsedFor	make boat
work	NotHasProperty	fun	work	NotHasProperty	bore
shark	AtLocation	any ocean	shark	AtLocation	coral reef
chicken	CapableOf	cross road	chicken	CapableOf	fly
chicken	CapableOf	produce egg	chicken	CapableOf	fly
drink	HasPrerequisite	open your mouth	drink	HasPrerequisite	buy beer
window	UsedFor	look outside	window	UsedFor	let light in
football	IsA	game	football	IsA	popular sport
car	HasA	seat	car	HasA	seatbelt
car	HasA	four wheel	car	HasA	seatbelt
car	CapableOf	slow down	car	CapableOf	roll down street
love	CausesDesire	propose to woman	love	CausesDesire	copulate
knife	CapableOf	spread butter	knife	CapableOf	hurt person
knife	CapableOf	spread peanut butter	knife	CapableOf	hurt person

Head entity	Realtion	Tail entity	Head entity	Realtion	Tail entity
knife	UsedFor	stab	knife	UsedFor	slice meat
music	CreatedBy	composer	music	CreatedBy	write
music	HasProperty	relax	music	HasProperty	important to human
ball	IsA	toy	ball	IsA	throw
do housework	Causes	have clean house	do housework	Causes	sweat
bird	AtLocation	sky	bird	AtLocation	in tree
bird	AtLocation	roof	bird	AtLocation	in tree
bird	AtLocation	air	bird	AtLocation	in tree
bird	CapableOf	sing	bird	CapableOf	lay egg
feather	UsedFor	tickle someone	feather	UsedFor	tickle
teacher	CapableOf	school student	teacher	CapableOf	teach
teacher	CapableOf	help student	teacher	CapableOf	teach
pool	UsedFor	get out of heat	pool	UsedFor	dunk
human	CapableOf	die only once	human	CapableOf	commit genocide
human	AtLocation	school	human	AtLocation	hospital
human	AtLocation	love	human	AtLocation	hospital
human	AtLocation	country	human	AtLocation	hospital
human	AtLocation	home	human	AtLocation	hospital
human	AtLocation	park	human	AtLocation	hospital
human	AtLocation	workplace	human	AtLocation	hospital
plant	HasA	leave	plant	HasA	stem
read	HasSubevent	turn page	read	HasSubevent	learn something
pilot	CapableOf	land plane	pilot	CapableOf	fly aiplane
pilot	CapableOf	land airplane	pilot	CapableOf	fly aiplane
person	CapableOf	voice opinion	person	CapableOf	die
person	CapableOf	love	person	CapableOf	die
person	CapableOf	catch cold	person	CapableOf	die
person	CapableOf	cross street	person	CapableOf	die
person	Desires	clothe	person	Desires	love
person	CapableOf	wind clock	person	CapableOf	die
person	CapableOf	taste food	person	CapableOf	die
person	CapableOf	thank another person	person	CapableOf	die
person	CapableOf	captain ship	person	CapableOf	die
person	CapableOf	water plant	person	CapableOf	die
person	CapableOf	believe in god	person	CapableOf	die
person	Desires	dance	person	Desires	love
person	CapableOf	laugh at joke	person	CapableOf	die
person	CapableOf	talk to each other	person	CapableOf	die
person	CapableOf	thank god	person	CapableOf	die
person	CapableOf	sail boat	person	CapableOf	die
person	CapableOf	pay bill	person	CapableOf	die
person	CapableOf	shoulder burden	person	CapableOf	die
person	Desires	laugh	person	Desires	love
person	Desires	feel important	person	Desires	love
person	CapableOf	board plane	person	CapableOf	die
doctor	CapableOf	help sick person	doctor	CapableOf	treat seriously ill person
boy	CapableOf	date girl	boy	CapableOf	like boy
boy	CapableOf	kiss girl	boy	CapableOf	like boy
boy	IsA	young man	boy	IsA	male kid of his parent
have sex	Causes	baby	have sex	Causes	pregnancy
have sex	HasSubevent	sweat	have sex	HasSubevent	make love
match	CapableOf	light candle	match	CapableOf	ignite
foot	AtLocation	desk	foot	AtLocation	toe
exercise	HasPrerequisite	energy	exercise	HasPrerequisite	go for run
magician	CapableOf	fool audience	magician	CapableOf	do magic
rock	IsA	music	rock	IsA	rock with smooth edge
cook	CapableOf	prepare meal	cook	CapableOf	bread chicken
student	CapableOf	fail test	student	CapableOf	read
student	CapableOf	master subject	student	CapableOf	read
fruit	HasProperty	edible	fruit	HasProperty	high in calorie
boat	AtLocation	water	boat	AtLocation	sea
boat	IsA	on water	boat	IsA	usually
rain	IsA	water	rain	IsA	form of weather
host	CapableOf	welcome guest	host	CapableOf	tape television show
ride horse	HasSubevent	fall off	ride horse	HasSubevent	ride pony
horse	HasProperty	brown	horse	HasProperty	yellow
horse	ReceivesAction	ride	horse	ReceivesAction	train to jump high fence
horse	CapableOf	carry person	horse	CapableOf	jump
horse	AtLocation	race track	horse	AtLocation	ranch
earth	HasA	one moon	earth	HasA	more water than land
earth	NotIsA	perfect sphere	earth	NotIsA	planet
someone	AtLocation	museum	someone	AtLocation	party
someone	AtLocation	post office	someone	AtLocation	party
someone	AtLocation	zoo	someone	AtLocation	party
someone	AtLocation	lake	someone	AtLocation	party
someone	AtLocation	shop	someone	AtLocation	party
bicycle	IsA	two wheel vehicle	bicycle	IsA	method of transportation
go to bed	HasPrerequisite	turn off light	go to bed	HasPrerequisite	go to your bed
go to bed	HasPrerequisite	turn out light	go to bed	HasPrerequisite	go to your bed
play game	HasSubevent	fun	play game	HasSubevent	run
elephant	AtLocation	circus	elephant	AtLocation	jungle

Head entity	Realtion	Tail entity	Head entity	Realtion	Tail entity
go to get haircut	Causes	your hair will be short	go to get haircut	Causes	short hair
play	MotivatedByGoal	have some fun	play	MotivatedByGoal	win
play	Causes	fun	play	Causes	entertainment
man	CapableOf	father child	man	CapableOf	breathe
man	CapableOf	date woman	man	CapableOf	breathe
man	Desires	woman	man	Desires	love
fall	IsA	season	fall	IsA	activity
stop your bicycle	HasSubevent	apply brake	stop your bicycle	HasSubevent	brake
kill	HasProperty	wrong	kill	HasProperty	fun
finger	CapableOf	push button	finger	CapableOf	scratch
star	AtLocation	night sky	star	AtLocation	orbit
goldfish	IsA	carp	goldfish	IsA	common pet
table	UsedFor	put thing on	table	UsedFor	stand on
use computer	HasSubevent	type on keyboard	use computer	HasSubevent	play video game
watch movie	HasPrerequisite	buy ticket	watch movie	HasPrerequisite	go to movie
canada	HasProperty	north of unite state	canada	HasProperty	very cold
home	RelatedTo	family	home	RelatedTo	house
neighbour	AtLocation	door	neighbour	AtLocation	neighbor
go for drive	HasPrerequisite	get car	go for drive	HasPrerequisite	get key
something	AtLocation	something else	something	AtLocation	box
something	AtLocation	school	something	AtLocation	box
something	AtLocation	store	something	AtLocation	box
something	AtLocation	beach	something	AtLocation	box
something	AtLocation	tree	something	AtLocation	box
something	AtLocation	mall	something	AtLocation	box
something	AtLocation	sea	something	AtLocation	box
something	AtLocation	refrigerator	something	AtLocation	box
girl	IsA	hold puppy	girl	IsA	boy
pretend	HasPrerequisite	imagine	pretend	HasPrerequisite	dummy
potato	AtLocation	restaurant	potato	AtLocation	pizza
city	AtLocation	county	city	AtLocation	new york
frisbee	HasProperty	round	frisbee	HasProperty	have fun
go for haircut	HasPrerequisite	find barber	go for haircut	HasPrerequisite	go to barber shop
comb	CapableOf	part hair	comb	CapableOf	remove tangle from hair
comb	UsedFor	style hair	comb	UsedFor	tidy person hair
stapler	AtLocation	your desk	stapler	AtLocation	drawer
chess board	HasA	64 square	chess board	HasA	rook
classroom	IsA	in school	classroom	IsA	place
gravity	IsA	force	gravity	IsA	direction opposite pull of gravity
day	HasProperty	bright	day	HasProperty	late
paint	CapableOf	coat wall	paint	CapableOf	cover
canvas	UsedFor	paint on	canvas	UsedFor	paint picture
have rest	MotivatedByGoal	you be very tire	have rest	MotivatedByGoal	relax
get job	HasSubevent	interview	get job	HasSubevent	work
lizard	AtLocation	dessert	lizard	AtLocation	forest
lizard	AtLocation	bush	lizard	AtLocation	forest
pencil	UsedFor	write something on paper	pencil	UsedFor	write letter
plane	CapableOf	arrive at airport	plane	CapableOf	runway
woman	HasA	baby	woman	HasA	penis
woman	CapableOf	chair committee	woman	CapableOf	love
woman	CapableOf	mother child	woman	CapableOf	love
woman	CapableOf	wear dress	woman	CapableOf	love
drive	HasSubevent	listen to radio	drive	HasSubevent	turn key
coffee	HasProperty	serve hot	coffee	HasProperty	sweet
slide	AtLocation	park	slide	AtLocation	playground equipment
basketball	HasProperty	round	basketball	HasProperty	fill with air
agree with someone	HasSubevent	nod	agree with someone	HasSubevent	nod head
sand	AtLocation	desert	sand	AtLocation	find on beach
tongue	CapableOf	taste food	tongue	CapableOf	kiss
play piano	HasPrerequisite	take lesson	play piano	HasPrerequisite	practice piano
mouse	AtLocation	laboratory	mouse	AtLocation	build
crab	AtLocation	sand	crab	AtLocation	saltwater
blanket	UsedFor	sleep	blanket	UsedFor	keep warm
farmer	CapableOf	farm land	farmer	CapableOf	plant crop
air	IsA	gas	air	IsA	light than air
brain	IsA	head	brain	IsA	complex organ
thief	CapableOf	case house	thief	CapableOf	steal from car
thief	CapableOf	case joint	thief	CapableOf	steal from car
neighbor	AtLocation	door	neighbor	AtLocation	neighbor house
read magazine	HasSubevent	turn page	read magazine	HasSubevent	read
babysitter	CapableOf	mind baby	babysitter	CapableOf	dress child
see your favorite show	Causes	laugh	see your favorite show	Causes	enjoyment
propose to woman	MotivatedByGoal	you love her	propose to woman	MotivatedByGoal	marriage
reproduce	HasPrerequisite	find mate	reproduce	HasPrerequisite	procreate
reproduce	Causes	child	reproduce	Causes	make baby
internet	UsedFor	research	internet	UsedFor	get information
cold	CausesDesire	light fire	cold	CausesDesire	get warm
fan	UsedFor	move air	fan	UsedFor	cool person on hot day
artist	CapableOf	paint portrait	artist	CapableOf	paint picture
bathe	HasSubevent	use soap	bathe	HasSubevent	get naked
soap	UsedFor	wash yourself	soap	UsedFor	wash dirt from between your toe

Head entity	Realtion	Tail entity	Head entity	Realtion	Tail entity
soap	UsedFor	clean something	soap	UsedFor	wash dirt from between your toe
fly in airplane	HasPrerequisite	buy ticket	fly in airplane	HasPrerequisite	become pilot
get something	HasPrerequisite	ask for it	get something	HasPrerequisite	go to store
go for swim	HasSubevent	drown	go for swim	HasSubevent	get in water
chew your food	Causes	good digestion	chew your food	Causes	chew
play guitar	HasSubevent	sing	play guitar	HasSubevent	make music
lawyer	CapableOf	object in court	lawyer	CapableOf	object to issue
lawyer	CapableOf	settle lawsuit	lawyer	CapableOf	object to issue
hummingbird	CapableOf	hover	hummingbird	CapableOf	fly
star trek	IsA	popular television show	star trek	IsA	television show
stage	UsedFor	play	stage	UsedFor	do performance
play poker	HasSubevent	bluff	play poker	HasSubevent	cheat
salt	UsedFor	melt ice	salt	UsedFor	flavor water
atheist	CapableOf	doubt existence of god	atheist	CapableOf	believe in god
detective	CapableOf	piece together clue	detective	CapableOf	catch criminal
alcohol	CapableOf	cloud judgement	alcohol	CapableOf	get you drunk
some car	HasProperty	expensive	some car	HasProperty	yellow
your neighbor	AtLocation	door	your neighbor	AtLocation	next door
remember phone number	HasPrerequisite	repeat it to yourself	remember phone number	HasPrerequisite	commit to memory
moon	AtLocation	space	moon	AtLocation	orbit
oxygen	IsA	gas	oxygen	IsA	atom
train	CapableOf	arrive late	train	CapableOf	run
submarine	IsA	ship	submarine	IsA	military submarine
move car	Causes	accident	move car	Causes	drive it