ReadE: Learning Relation-Dependent Entity Representation for Knowledge Graph Completion

Anonymous ACL submission

Abstract

Existing knowledge graph embedding methods that adopt powerful graph neural networks try to aggregate well-preserved neighborhood information into the entity representation. However, they represent each entity solely with a relation-irrespective representation which contains the entire miscellaneous neighborhood information, regardless of the variance of emphatic semantics required by different relations in predicting the missing entities. To tackle this problem, we propose ReadE, a method to learn relation-dependent entity representation, of which the neighborhood information is selectively aggregated and emphasized by varied relations types. First, we propose a relation-controlled gating mechanism targeting on utilizing the relation to control the information flow from neighbors in the aggregation step of the graph neural network. Second, we propose a well-designed contrastive learning method with mixing both relation-level and entity-level negative samples to enhance semantics preserved in our relation-dependent GNN-based representations. Experiments on three benchmarks show that our proposed model outperforms all strong baselines. The code will be made open-sourced on Github.

1 Introduction

Knowledge graph (KG) is a semantic network and can be used to represent the relations of different entities in the real world. Due to the existence of a huge amount of potential facts, existing KGs, like NELL (Carlson et al., 2010) and YAGO3 (Mahdisoltani et al., 2015), mostly face the problem of completing the missing relations, which is known as the knowledge graph completion (KGC) task. In this work, we mainly focus on the task of how to predict the missing entity in incomplete triplets like $< entity, relation, ? >$.

To complete the KG, a fundamental task is to learn informative and meaningful representations for the entities and relations in KG, based on which the missing links can be predicted. Given a triplet $< e_1, r, e_2 >$, TransE (Bordes et al., 2013) proposed to learn the representations that satisfy the property of translation invariance $\boldsymbol{e}_1 + \boldsymbol{r} \approx \boldsymbol{e}_2$. To increase the model's representational ability, in ConvE (Dettmers et al., 2018), a multi-layer convolution network is used to predict missing entities. However, all these methods process each triplet independently, ignoring the neighborhood information inherent in the graph structure. To address this, many methods adopt various kinds of graph neural networks (GNNs) to aggregate the neighbor's information into the entity representation (Nathani et al., 2019; Shang et al., 2019). For example, GAATs (Wang et al., 2020) introduce a graph attenuated attention mechanism to consider n-hop neighbors and assign different weights for different relation paths to well-preserve the neighborhood information. KE-GCN (Yu et al., 2021a) adopts the graph convolutional network (GCN) to model the homogeneous topology information that exists in a KG. Similarly, HRAN (Li et al., 2021) introduces a heterogeneous GCN to model heterogeneous relation feature.

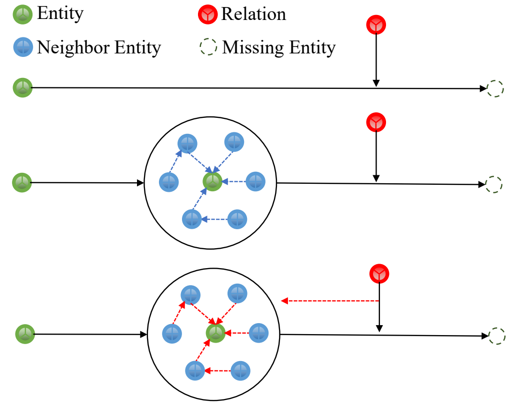


Figure 1: A simple illustration of different kinds of KGE methods. From top to bottom are conventional methods, existing GNN-based approaches to learn relation-irrespective representations, and our model to learn relation-dependent representations.

Intuitively, as pointed out in TransR (Lin et al., 2015), an entity could play different roles. Hence, the representation of an entity that is learned by GNN-based approaches may contain miscellaneous neighborhood information of many aspects derived from its neighbor entity nodes. For example, when *MichaelJordan* is working for *ChicagoBulls*, he has a teammate *ScottiePippen* (*i.e.*, a neighbor), who was born in *Hamburg*. Therefore, the representation of *MichaelJordan* learned by existing graph neural networks intends to contain information about *ScottiePippen*'s birthplace and employment information simultaneously. When an incomplete triplet $< MichaelJordan, EmploymentCompany, ? >$ is given, if the information of *ScottiePippen* that is related to the relation *EmploymentCompany* can be emphasized and aggregated into the representation of *MichaelJordan*, it would be easier to predict the ground-truth missing entity *ChicagoBulls*. Therefore, it is important for every entity to have a representation that is dependent on its corresponding concrete relation. That is, when interacting with different relations to predict the miss one, an entity needs to show selective neighborhood information according to the relation it connects with. However, existing methods only learn a relation-irrespective representation for an entity, irrespective of the exact relations they interact with. For these relation-irrespective entity representations, obviously, different aspects of neighborhood information cannot be shown when interacting with different relations in predicting missing entities.

In this paper, we propose ReadE, a method to learn **Rela**tion-**d**ependent **E**ntity representations. Fig 1 visually illustrates the difference between our proposed model and previous methods. In our proposed method, the representation of an entity can vary according to the relation that is interacted with. To this end, we first propose a relation-controlled gating mechanism that is used to control which and how much information from neighbors can flow into the interested entity's representation during the aggregation step. Since a good relation representation can make the relation-controlled gating mechanism work better, in contrast to previous methods, a similarity-preserving relation representation is learned for every relation through GCN, hoping that similar relations (*e.g.*, *PlaceOfBorn* and *PlaceOfResidence*) in the graph can share similar representations, capturing the correlation among different relations. Moreover, we further propose to use contrastive learning to enhance the semantic information in our relation-dependent entity representation, in which a novel two-level generation process of negative samples is proposed. Extensive experiments are conducted on three benchmarks for the knowledge graph completion task. The experiments show that our ReadE outperforms all strong baselines and further analyses verify the validity of each proposed component.

2 Related Work

Nowadays, knowledge graph embedding (KGE) methods play an important role in KGC. Given a triplet $< e_1, r, e_2 >$, TransE (Bordes et al., 2013) learns the representation of the entity and relation according to the translation-based constraint of $\boldsymbol{e}_1 + \boldsymbol{r} \approx \boldsymbol{e}_2$. Later, TransH (Wang et al., 2014), TransR (Lin et al., 2015), and TransD (Ji et al., 2015) extend the translation-based constraint to model more complex features. To further learn more expressive representation, ConvE (Dettmers et al., 2018) adopts multi-layer CNN architecture to capture the deeper correlation between e_1 and r. Then, ConvKB (Nguyen et al., 2018) further extends ConvE to consider correlation between the entire triplet (e_1, r, e_2). InteractE (Vashishth et al., 2020) introduces more types of interactions between entity and relation in ConvE. For more details, we refer interested readers to some surveys (Wang et al., 2017; Nguyen, 2020).

However, these methods process each triplet independently, ignoring the neighborhood information inherent in the graph structure of a given entity. To address this, Nathani et al. (2019) adopt the graph attention network to aggregate the information from neighbors to obtain a meaningful entity representation. Similarly, KE-GCN (Yu et al., 2021a) adopts the GCNs to simultaneously model the entities and relations and then capture heterogeneous relations in the knowledge graph. SACN (Shang et al., 2019) adopts the weighted GCN, which assigns each relation a trainable weight and then aggregates information from neighbors according to the connected relation. Further, COMPGCN (Vashishth et al., 2019) targets at the directed multi-relational KG, and proposes to systematically leverage entity-relation composition operations via GCN-based approach. GAATs (Wang et al., 2020) argues that different relation paths in KG should be assigned with different weights and integrate an attenuated attention mechanism to better preserve the neighborhood information. Later, HRAN (Li et al., 2021) divides the KG into sub-graph levels, where each sub-graph contains all the entities but only 1 relation, to capture the heterogeneous relation features.

Another approach to KGE is to adopt the Transformer architecture (Vaswani et al., 2017), *e.g.*, KG-BERT (Yao et al., 2019), StAR (Wang et al., 2021), and HittER (Chen et al., 2021). These models adopt the deep Transformer architecture to learn a more meaningful representation and then advance the KGC, but they are usually urgent for huge computing resources.

Our paper belongs to the category that considers neighborhood information. It can be concluded that the existing methods represent each entity solely with a relation-irrespective representation which contains the entire miscellaneous neighborhood information. Different from them, given incomplete triplets like $< entity, relation, ? >$, we use the $relation$ as the guidance to selectively aggregate the neighborhood information into the entity representation. From this perspective, the existing GNN-based representations are regarded as relation-irrespective, while our representation is relation-dependent.

3 Preliminary

Due to the strong ability to learn commonalities among adjacent nodes for graph-structured data, graph neural networks (GNN) have been widely used to learn the entity representations of knowledge graphs in recent years (Nathani et al., 2019; Shang et al., 2019; Li et al., 2021). The GNN-based models generally share the common architecture of using a GNN to learn the entity representation and then applying a score function to evaluate the matching degree of a triplet <head entity, relation, tail entity>. Because of the similarity among these methods, here we take the SACN (Shang et al., 2019) as an example to illustrate the basic principles behind the GNN-based entity representation learning methods.

By viewing the KG as a entity graph G_e, in which each node and edge represents an entity and relation, respectively, SACN applies a L-layer weighted graph convolutional network onto graph G_e to obtain entity representations

$$\boldsymbol{z}_i^l = \sigma\left(\sum_{j \in \mathcal{N}_e(i)} \alpha_{i,j} \boldsymbol{z}_j^{l-1} \boldsymbol{W}^{l-1} + \boldsymbol{z}_i^{l-1} \boldsymbol{W}^{l-1} \right), \quad (1)$$

where $\ell = 1, 2, \cdots L$ denotes the ℓ-th layer of GNN; $\mathcal{N}_e(i)$ represents the neighbors of entity i in graph G_e; $\boldsymbol{z}_i^\ell$ denotes the embedding of i-th entity e_i obtained at the ℓ-th layer, with the initial embeding $\boldsymbol{z}_i^0 \in \mathbb{R}^{d_e}$ initialized from random Gaussian noise; $\boldsymbol{W}^l \in \mathbb{R}^{d_e \times d_e}$ is the network parameter at $(\ell - 1)$-th layer; the coefficient $\alpha_{i,j}$ is used to control the interaction strength between node i and j; and $\sigma(\cdot)$ is the sigmoid activation function. $\boldsymbol{z}_i^L$ from the L-th layer is then used to represent the final embedding of the i-th entity e_i, that is,

$$\boldsymbol{z}_i = \boldsymbol{z}_i^L. \quad (2)$$

Besides the embedding $\boldsymbol{z}_i$, SACN also learns an embedding for every relation r. For the k-th relation r_k, its embedding $\boldsymbol{h}_k \in \mathbb{R}^{d_e}$ is directly initialized from a random Gaussian noise.

Using the entity embeddings $\boldsymbol{z}_i$ and relation embeddings $\boldsymbol{h}$ obtained above, for a given triplet $< e_i, r_k, e_j >$, the SACN evaluates a matching score for it with a scoring function of the form

$$\varphi(\boldsymbol{z}_i, \boldsymbol{h}_k, \boldsymbol{z}_j) = CNN\left([\boldsymbol{z}_i; \boldsymbol{h}_k]\right) \boldsymbol{W}^c \boldsymbol{z}_j^T, \quad (3)$$

where $CNN(\cdot)$ denotes a convolutional network applied to a $2 \times d_e$ matrix $[\boldsymbol{z_i}; \boldsymbol{h}_k]$. The model will compute the probability that the given triplet $< e_i, r_k, e_j >$ is true as

$$p(e_i, r_k, e_j) = \sigma(\varphi(\boldsymbol{z}_i, \boldsymbol{h}_k, \boldsymbol{z}_j)). \quad (4)$$

Given a training dataset containing both of true and false triplets, the model parameters and initial embeddings can be optimized by minimizing the following cross-entropy loss

$$\mathcal{L}_c = \frac{-1}{N} \sum_{n=1}^{N} (y_n \log p_n + (1 - y_n) \log(1 - p_n)) \quad (5)$$

where p_n denotes the probability of truth for the n-th triplet computed according to (4); and y_n is the ground-truth label, which is 1 for true triplet and 0 otherwise.

4 Methodologies

In this section, we propose our ReadE. First, we present how to learn relation-dependent representations through a relation-controlled gating mechanism, then introduce a novel contrastive learning

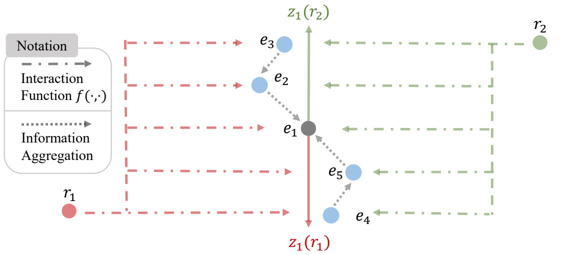


Figure 2: The overall framework of our designed relation-dependent entity representation learning method.

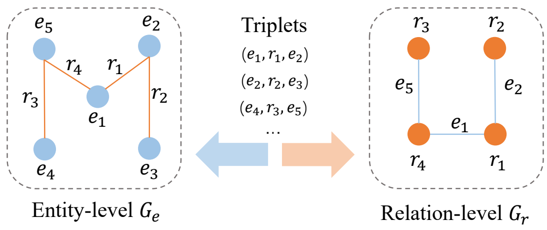


Figure 3: The illustration of our graph construction methods of G_r and G_e.

method with mixing both relation-level and entity-level negative samples to enhance the entities' semantic information.

4.1 Relation-Dependent Entity Representation Learning

Existing methods mainly focus on how to learn good representations for the entities and relations so that the relevance among the entities and relations in true triplets can be retained as much as possible. However, in all of these existing methods, the learned representation of an entity is never dependent on the relations, that is, the representation maintains one appearance under different relations. However, no matter the problem is to predict the tail entity given the head entity and relation $< e_i, r_k, ? >$, or to predict the head entity given the tail entity and relation $<?, r_k, e_j >$, the relation is always available. Thus, if we learn for every entity a collection of representations, with each corresponding to a relation, when facing the entity prediction task $< e_i, r_k, ? >$ or $<?, r_k, e_j >$, we can always choose to use the entity representation under the specific relation r_k. To the convenience of presentation, in the following, we denote the representation of i-th entity e_i under relation r as $\boldsymbol{z}_i(r)$.

The relation-dependent entity representation under relation r can be learned with a GNN as

$$\boldsymbol{z}_i^l(r) = \frac{1}{|\mathcal{N}_e(i)|} \sum_{j \in \mathcal{N}_e(i)} f(\boldsymbol{h}_r, \boldsymbol{z}_j^{l-1}(r)) \boldsymbol{W}^{l-1} + f(\boldsymbol{h}_r, \boldsymbol{z}_i^{l-1}(r)) \boldsymbol{W}^{l-1}. \quad (6)$$

Here, $f(\cdot, \cdot)$ is the interaction function between the entity and relation and is designed as

$$f(\boldsymbol{h}_r, \boldsymbol{z}_i^{l-1}(r)) = \sigma\Big(\boldsymbol{W}^f \boldsymbol{h}_r + \boldsymbol{g}^f\Big) \odot \boldsymbol{z}_i^{l-1}(r), \quad (7)$$

where $\boldsymbol{W}^f \in \mathbb{R}^{d_e \times d_r}$, $\boldsymbol{g}^f \in \mathbb{R}^{d_e}$ are parameters to be learned, $\odot$ is the feature-wise product. The final entity representation $\boldsymbol{z}_i(r)$ is obtained by applying the sigmoid function σ to the output at the last layer, *i.e.*, $\boldsymbol{z}_i(r) = \sigma(\boldsymbol{z}_i^L(r))$. The function $f(\cdot, \cdot)$ plays a role of relation-controlled gate that can determine which dimension's information in the entity representation $\boldsymbol{z}_i^{\ell-1}$ can be flowed into neighboring nodes. If the relevance between the relation and an entity is weak, the $\sigma(\cdot)$ function will output a value close to zero, cutting off the information flowing into to the entity's neighbors.

The reason why we design this relation-controlled gate function is that KGs are usually densely connected (Lovelace et al., 2021), making a GCN-based encoder prone to aggregate from its neighbors the irrelevant information *w.r.t.* the considered relation. Thus, as illustrated in Fig 2, as aggregating the information from neighbors, we first let the relation control which and how much information can flow into the interested entity's representation, making the entity have different representations under different relations.

Similarity-preserving Relation Representation Learning The relation dependence in the proposed entity representations is achieved by incorporating the relation representations $\boldsymbol{h}_k$ into the entities' representation updating process through a gating mechanism. In practice, different relations are related, rather than isolated, to each other. For example, in KG, the relation *PlaceOfBorn* and *PlaceOfResidence* are both related to the city entity, suggesting they should share some common semantic information in their representations. However, we cannot directly model such kind of similarities on the existing entity graph G_e, resulting in that the relation representations used in the gating function (7) fails to contain sufficient relevant information between different relations. To have the relation representations to reflect this kind of similarities, we propose to construct a relational graph G_r from the KG by representing every relation as a node and adding an edge between two relations if they connect to a common entity, as illustrated in Fig 3. With the relation graph G_r, we can now apply the graph neural networks (*e.g.*, GCN) on the graph to obtain relation representations

$$\boldsymbol{h}_r^l = \sigma\left(\sum_{j\in\mathcal{N}_r(r)} \boldsymbol{h}_j^{l-1}\boldsymbol{W}_r^{l-1} + \boldsymbol{h}_r^{l-1}\boldsymbol{W}_r^{l-1}\right), \quad (8)$$

where $\ell = 1, 2, \cdots, L'$ denotes the ℓ-th layer of GCN; the initial embedding $\boldsymbol{h}_r^0$ is initialized by random Gaussian noise; $\mathcal{N}_r(\cdot)$ denotes the set of the neighbors of relation r in G_r; and $\boldsymbol{W}_r^l \in \mathbb{R}^{d_r\times d_r}$ is the GCN parameter. We set the output $\boldsymbol{h}_r^{L'}$ from the last layer as the final relation representation, that is,

$$\boldsymbol{h}_r = \boldsymbol{h}_r^{L'}. \quad (9)$$

Thanks to the message-passing process during the learning, the representation of a relation is not isolated anymore, but is related to other relations that share common entities. In this way, the common information of different relations or their similarity information can be manifested in the learned representations. By substituting the similarity-preserved relation representation (9) into entity representation updating equation (6), the final relation-dependent entity representation updating method is obtained.

4.2 Enhancing Semantics of Entity Representation with Contrastive Learning

The link prediction task is to predict the missing head or tail entity given the other two components. Thus, similar to the classification tasks in images and texts, if more semantic information of entities are preserved in their representations, better prediction performance can be expected. Technically, contrastive learning can be understood as finding pairs of positive and negative instances and then trying to reduce the distance between positive pairs while enlarging that between negative ones under different contrast losses. Among them, the NT-Xent (Chen et al., 2020) contrast loss below is used most widely

$$\mathcal{L} = -\log\frac{\mathcal{D}(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_i^{(2)})}{\mathcal{D}(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_i^{(2)}) + \sum_{j\neq i, m=1,2}\mathcal{D}(\boldsymbol{u}_i^{(1)}, \boldsymbol{u}_j^{(m)})},$$

where $\boldsymbol{u}_i^{(m)}$ represents the m-th view of the i-th instance. Different views from the same instances are generally treated as positive pairs, while views from different instances are considered as negative pairs. The key of using contrastive learning lies at how to find effective positive and negative pairs, which can determine whether semantic information can be well preserved in the representations. For images, both of the positive and negative pairs can be easily obtained by applying transformations to the same or different images. However, for graphs, especially for knowledge graphs that contain the additional information of relation, generating effective positive and negative pairs is not that straightforward at all.

To generate positive pairs, inspired by the works that apply self-supervised learning on general graphs (Velickovic et al., 2019; Xia et al., 2021; Yu et al., 2021b), we perturb the knowledge graph by randomly dropping some nodes and edges and then apply the aforementioned methods on the perturbed graph to obtain the entities' representations $\boldsymbol{z}_i'$. Then, the representations $\boldsymbol{z}_i$ and $\boldsymbol{z}_i'$ can be viewed as a positive pair. For convenience of presentation, the two representations $\boldsymbol{z}_i$ and $\boldsymbol{z}_i'$ are deemed as two views of entity i, and are denoted as $\boldsymbol{z}_i^{(1)}$ and $\boldsymbol{z}_i^{(2)}$. The concrete steps to perturb the KG are described in the Appendix C.

As for the generation of negative pairs, a common method is to treat views of other entities as negative samples. However, in order to learn more meaningful semantic information in KG, we suggest to collect negative samples in two different levels, *i.e.*, the relation level and the entity level.

Relation-Level Negative Samples For a relation-dependent entity representation $\boldsymbol{z}_i(r)$, we hope it can retain discriminative semantic information of entity i under the specific relation of r. To strength the objective that the semantic information contained in $\boldsymbol{z}_i(r)$ is exclusive to the relation r, we propose to generate negative samples under the same entity by using different relations r' with $r' \neq r$. Specifically, for the representation of entity e_i under the relation r, *i.e.*, $\boldsymbol{z}_i(r)$, its relation-level negative samples is defined to be from the following set

$$\mathcal{Z}_i^{neg}(r) = \left\{\boldsymbol{z}_i^{(1)}(r'), \boldsymbol{z}_i^{(2)}(r')\,\middle|\, r' \neq r\right\}. \quad (10)$$

Entity-Level Negative Samples For an entity representation $\boldsymbol{z}_i(r)$, in addition to include exclusive semantic information comparing to entity representations under other relations $\boldsymbol{z}_i(r')$ with $r' \neq r$, it should also contain exclusive semantic information when comparing with other entities. Therefore, we define the entity-level negative samples of $\boldsymbol{z}_i(r)$ as

$$\widetilde{\mathcal{Z}}_i^{neg}(r) = \left\{ \boldsymbol{z}_j^{(1)}(r), \boldsymbol{z}_j^{(2)}(r) \middle| j \neq i \right\}, \quad (11)$$

where we require the relation in other entities to be the same as the considered entity. In the implementation, the entity j can just be the other entities from the same mini-batch.

With the two negative sample sets, we can define the final contrastive learning loss as

$$\ell_i^{(1)} = -\log \frac{\mathcal{D}_{pos}}{\mathcal{D}_{pos} + \sum_{\boldsymbol{u} \in \mathcal{Z}_i(r)} \mathcal{D}(\boldsymbol{z}_i^{(1)}(r), \boldsymbol{u})}, \quad (12)$$

where $\mathcal{Z}_i(r) \triangleq \mathcal{Z}_i^{neg}(r) \cup \widetilde{\mathcal{Z}}_i^{neg}(r)$; and $\mathcal{D}_{pos} \triangleq \mathcal{D}(\boldsymbol{z}_i^{(1)}(r), \boldsymbol{z}_i^{(2)}(r))$. Here, $\mathcal{D}(\boldsymbol{z}_i^{(1)}(k), \boldsymbol{z}_i^{(2)}(k))$ is calculated as

$$\mathcal{D}(\boldsymbol{z}_i^{(1)}(k), \boldsymbol{z}_i^{(2)}(k)) = e^{sim(\boldsymbol{z}_i^{(1)}(k), \boldsymbol{z}_i^{(2)}(k))/\tau}, \quad (13)$$

where $sim(\cdot, \cdot)$ denotes the cosine similarity between vectors, and τ is a temperature parameter controlling the concentration level of the distribution (Hinton et al., 2015). By averaging over a mini-batch of size N, the final contrastive loss $\mathcal{L}_{cl}$ is

$$\mathcal{L}_{cl} = \frac{1}{2N} \sum_{i=1}^{N} (\ell_i^{(1)} + \ell_i^{(2)}). \quad (14)$$

By minimizing $\mathcal{L}_{cl}$ with both the relation-level and entity-level negative samples, our ReadE can learn a entity representation preserving more meaningful semantics. Finally, we unify the objective of the KGC task and the contrastive learning as:

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_{cl}, \quad (15)$$

where λ is a hyper-parameter used to control the trade-off between the loss function.

5 Experiments

5.1 Datasets, Evaluation and Baselines

Datasets We evaluate the proposed ReadE model on three benchmark datasets, including FB15k-237 (Toutanova and Chen, 2015), WN18RR (Bordes et al., 2013) and UMLS (Kok and Domingos, 2007). Details of the three datasets can be found in Appendix.

Evaluation Metrics We evaluate the performance of our ReadE model on the link prediction task, *i.e.*, predicting the missing entity. In the inference phase, given an incomplete triplet, our model takes all the entities as the candidates and outputs the probabilities over all the candidates. Then each candidate is re-ranked according to their probabilities to calculate the Mean rank (MR), Mean reciprocal rank (MRR), and Hits@N. MR is the average of the rankings of entities predicted correctly over all triplets while MRR targets at the average of reciprocal rankings. Hits@N denotes the ratio of those predicted correctly entities which are ranked in top-N. Also, We follow Shang et al. (2019) to use the filtered setting Bordes et al. (2013), which will filter out all valid triplets before ranking.

In addition, we follow Sun et al. (2020) to adopt the "RANDOM" protocol to handle the situation that the ground-truth triplets have the same scores as the negative triplets, which is caused by the float precision problem. Namely, the rankings of triplets with the same scores will be randomly determined.

Baselines We compare our model with following strong baselines: TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2018), R-GCN (Schlichtkrull et al., 2018), RotatE (Sun et al., 2019), SACN (Shang et al., 2019), COMPGCN (Vashishth et al., 2019), ATTH (Chami et al., 2020), InteractE (Vashishth et al., 2020), TorusE (Ebisu and Ichise, 2020), PairRE (Chao et al., 2021), HRAN (Li et al., 2021).

5.2 Experimental Results

The experimental results of our ReadE and the strong baselines on FB15k-237, WN18RR, and UMLS are shown in Table 1. From the table, the proposed ReadE outperforms the strongest baseline HRAN significantly, with relative MRR improvement of 4.5% and 2.3% on FB15K-237 and WN18RR, respectively. Among all the baselines, SACN is the most similar one to our model. SACN and our ReadE both utilize the Conv-TransE model to predict the missing entity, and the main difference is that SACN learns a unique representation for each entity while ReadE learns a relation-dependent entity representation instead. It can be seen that our model outperforms SACN by 5.4% and 4.3% in MRR on FB15K-237 and WN18RR respectively, showing the effectiveness of the relation-

Model	FB15k-237			WN18RR			UMLS	
	Hits			Hits			Hits	
	@10	@1	MRR	@10	@1	MRR	@10	MR
TransE	0.441	0.198	0.279	0.532	0.043	0.243	0.989	1.84
DistMult	0.446	0.199	0.281	0.504	0.412	0.444	0.846	5.52
ComplEx	0.450	0.194	0.278	0.530	0.409	0.449	0.967	2.59
ConvE	0.497	0.225	0.312	0.531	0.419	0.456	0.990*	**1.00***
ConvKB	0.421	0.155	0.243	0.520	0.400	0.430	—	—
R-GCN	0.300	0.100	0.164	0.207	0.080	0.123	—	—
RotatE	0.533	0.241	0.338	0.571	0.428	0.476	—	—
SACN	0.536	0.261	0.352	0.535	0.427	0.470	—	—
COMPGCN	0.535	0.264	0.355	0.546	0.443	0.479	—	—
ATTH	0.501	0.236	0.324	0.551	0.419	0.466	—	—
InteractE	0.535	0.263	0.354	0.528	—	0.463	—	—
TorusE	0.484	0.217	0.316	0.512	0.422	0.452	—	—
PairRE	0.544	0.256	0.351	—	—	—	—	—
HRAN	0.541	0.263	0.355	0.542	0.450	0.479	—	—
ReadE	**0.562**	**0.275**	**0.371**	**0.555**	**0.460**	**0.490**	**0.993**	1.43
Improvements	**3.3%**	**4.2%**	**4.5%**	**2.4%**	**2.2%**	**2.3%**	—	—

Table 1: Performances on FB15k-237, WN18RR, and UMLS datasets. The performances of ConvE on UMLS are taken from the author's Github and are marked with *.

dependent entity representation.

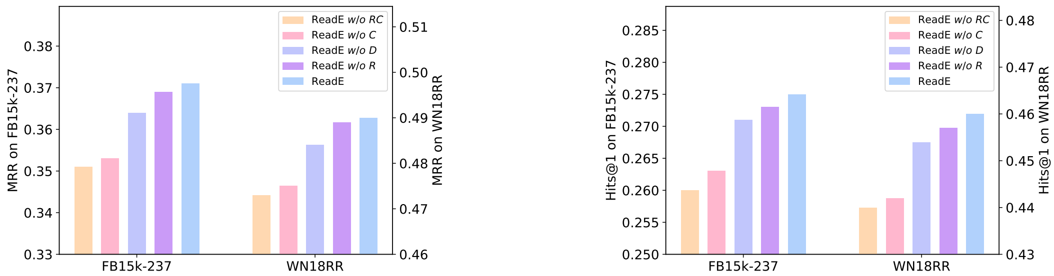
On UMLS, ReadE shows comparable performance with baselines. However, it is undeniable that ConvE outperforms our model on UMLS under the MR criterion. This may be due to the small size of UMLS, which leads to the over-fitting issue when injecting the graph structure information into the entity representation. However, On FB15k-237 and WN18RR with the more complex graph structure, our ReadE outperforms ConvE by 18.9% and 7.5% under the MRR criterion.

5.3 Impacts of Different Components

In this section, we give a deep insight into how much improvement different components contribute to the model performance. To do this, we evaluate the performance of variants of ReadE that exclude one or more components that have a large impact on the performance.

Specifically, three components included in ReadE are considered, and we follow our model's pipeline to describe the three components in turn: (1) Component C . It uses the relation to **C**ontrol the neighborhood information aggregation during the GCN-based encoding stage to generate the relation-dependent entity representation. Without it, every entity will be assigned a unique representation instead. (2) Component R. It means the similarity-preserving **R**elation representation learning component which obtains the relation representation by applying GCN on G_r. Without it, the relation representation degenerates the one ignoring its similarity information. (3) Component D. The contrastive learning component with **D**ouble levels of negative samples is designed to enhance the semantics of the relation-dependent entity representation. Dropping this component means that we remove the contrastive loss $\mathcal{L}_{cl}$. Please note that the D component is based on the C component, if we drop the C component, the D component will be dropped simultaneously. Based on the above-defined components, we propose four variants of ReadE: ReadE w/o R, ReadE w/o D, ReadE w/o C, ReadE w/o RC. The four variants are compared with the original ReadE on FB15k-237 and WN18RR and results are shown in Fig 4.

From the result, we can have the following observations. First, ReadE w/o C which removes the most basic component C will induce a significant performance drop when compared with the complete ReadE, suggesting the importance of taking the relation into account when learning the entity representation. Second, without using the proposed CL component (*i.e.*, ReadE w/o D), an immediate performance drop is observed on FB15k-237 and WN18RR, which demonstrates the necessity of utilizing the designed CL method to further improve our relation-dependent entity representation. Third, ReadE w/o C is better than ReadE w/o RC, demonstrating that even if we solely learn a unique relation-irrespective entity representation as previous methods do, improving the quality of the relation representation can still improve the performance. Also, ReadE w/o R works worse than ReadE, which indicates that similarity-preserving relation representations can better control the information aggregation from the neighborhood. Last but not least, if we remove all three components

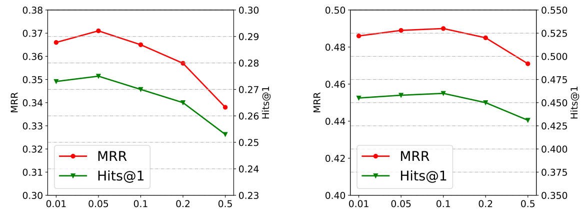


(a) MRR (b) Hits@1

Figure 4: Performances of variants of ReadE that exclude one or more components on FB15k-237 and WN18RR.

	FB15-237	WN18RR
w/o Entire $\mathcal{L}_{cl}$	0.364	0.484
+ Relation-Level	**0.369**	**0.488**
+ Entity-Level	0.366	0.486
+ Entire $\mathcal{L}_{cl}$	0.371	0.490

Table 2: MRR when using one of the relation-level and entity-level negative samples on FB15-237 and WN18RR.



(a) λ on FB15k-237 (b) λ on WN18RR

Figure 5: MRR and Hits@1 of ReadE under different values of λ on FB15k-237 and WN18RR.

(*i.e.*, ReadE w/o RC), the performance is poorest, confirming the validity of the proposed ReadE.

5.4 Impacts of Different Levels of Negative Samples

In this section, we evaluate the influence of relation-level and entity-level negative samples in the denominator of (14). MRR on FB15-237 and WN18RR datasets when using one of these two kinds of negative samples are shown in Table 2. Note that no matter which negative samples we use, the positive samples are unchanged and always be considered when calculating the contrastive loss.

We can see that the performances brought by CL with solely relation-level negative samples are more excellent than the ones brought by CL with solely entity-level negative samples on both FB15k-237 and WN18RR. In this paper, given an entity, CL with relation-level negative samples aims to increase the distances between different relation-dependent entity representations of it among different relations, which is consistent with our motivation of learning an entity representation of which semantics will vary depending on its relation. Therefore, it may explain why CL with relation-level negative samples achieves a greater result. Also, the model performs best if two levels of negative samples are considered together at the same time, indicating the credibility of the proposed CL method to enhance the entity's semantics.

5.5 Impacts of Parameter λ

In ReadE, we introduce the hyper-parameter λ , which controls the trade-off between the cross-entropy loss and the contrastive loss. In this section, we investigate the sensitivity of λ. We manually select the values of λ from $\{0.01, 0.05, 0.1, 0.2, 0.5\}$. MRR and Hits@1 w.r.t λ on FB15k-237 and WN18RR datasets are illustrated in Fig 5.

It is shown that as λ grows up, the performance of ReadE first increases and reaches the peak when $\lambda = 0.05$ and 0.1 on FB15k-237 and WN18RR respectively. Afterwards, if λ is larger, the improvement is neutralized and lost. This phenomenon shows that the performance is sensitive to the hyper-parameter λ. And in practice, we suggest that the loss weight for the contrastive loss can be set to $[0.01, 0.1]$ for exploiting the potentialities of the model.

6 Conclusion

In this paper, we proposed a novel knowledge graph embedding method, namely ReadE. In ReadE, we managed to introduce the relation-controlled gate mechanism to control the information flow in the aggregation step of the graph neural network, and thus obtained relation-dependent entity representations. Further, we proposed a contrastive learning method with both relation-level and entity-level negative samples for the particular purpose of enhancing the meaningful semantic information of entities' representations. Extensive experiments have shown that ReadE significantly outperformed existing baselines.

References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom Michael Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. PairRE: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4360–4369, Online. Association for Computational Linguistics.

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2021. Hitter: Hierarchical transformers for knowledge graph embeddings. In *EMNLP*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*.

Takuma Ebisu and Ryutaro Ichise. 2020. Generalized translation-based embedding of knowledge graph. *IEEE Transactions on Knowledge and Data Engineering*, 32:941–951.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. 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)*, pages 687–696, Beijing, China. Association for Computational Linguistics.

Stanley Kok and Pedro M. Domingos. 2007. Statistical predicate invention. In *ICML '07*.

Zhifei Li, Hai Liu, Zhaoli Zhang, Tingting Liu, and Neal Xiong. 2021. Learning knowledge graph embedding with heterogeneous relation attention networks. *IEEE transactions on neural networks and learning systems*, PP.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.

Justin Lovelace, Denis Newman-Griffis, Shikhar Vashishth, Jill Fain Lehman, and Carolyn Rosé. 2021. Robust knowledge graph completion with stacked convolutions and a student re-ranking network. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1016–1029, Online. Association for Computational Linguistics.

Farzaneh Mahdisoltani, Joanna Asia Biega, and Fabian M. Suchanek. 2015. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.

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

Dat Quoc Nguyen. 2020. A survey of embedding models of entities and relationships for knowledge graph completion. *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*.

M. Schlichtkrull, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC*.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *AAAI*.

Zhiqing Sun, Zhihong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *ArXiv*, abs/1902.10197.

Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. 2020. A re-evaluation of knowledge graph completion methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5516–5522, Online. Association for Computational Linguistics.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Pratim Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *AAAI*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. pages 5998–6008.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Lio', Yoshua Bengio, and R. Devon Hjelm. 2019. Deep graph infomax.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. *Proceedings of the Web Conference 2021*.

Quan Wang, Zhendong Mao, Biwu Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29:2724–2743.

Rui Wang, Bicheng Li, Shengwei Hu, Wenqian Du, and Min Zhang. 2020. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access*, 8:5212–5224.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *AAAI*.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *ArXiv*, abs/1909.03193.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823.

Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. 2021a. Knowledge embedding based graph convolutional network. *Proceedings of the Web Conference 2021*.

Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021b. Self-supervised multi-channel hypergraph convolutional network for social recommendation. *Proceedings of the Web Conference 2021*.

A More Details about Datasets

We evaluate the proposed ReadE model on three benchmark datasets, including FB15k-237 (Toutanova and Chen, 2015), WN18RR (Bordes et al., 2013) and UMLS (Kok and Domingos, 2007). Below shows the detailed descriptions of the three datasets, with their statistics summaries listed in Table 3.

1) FB15k-237 (Toutanova and Chen, 2015) contains the knowledge base relation triplets including real-world named entities and the relation. The FB15k-237 is the subset of the FB15K (Bordes et al., 2013), which is originally collected from Freebase. Different from the FB15K, the inverse relations are removed from FB15k-237.

2) WN18RR consists of English phrases and the corresponding semantic relations, which is derived from the WN18 (Bordes et al., 2013). Similar to FB15k-237, the inverse relations and the leaky data are removed from the WN18RR.

3) UMLS (Kok and Domingos, 2007), named Unified Medical Language System, is a medical KG dataset. It contains 135 medical entities and 46 semantic relations.

B Training Details

According to the performance observed on the validation set, we select the number of layers from {1, 2, 3, 4, 5}, the batch size from {4, 32, 128, 256, 1024}, the embedding size from {100, 200, 300}, the learning rate from {1e-3, 5e-4, 5e-5, 1e-5, 5e-6}, the dropout rate from {0.1, 0.3, 0.5}, the temperature τ from {0.1, 0.5, 1, 2, 10}, and the λ from {0.01, 0.05, 0.1, 0.2, 0.5}, with the best used for evaluation on the test set. All experiments are conducted on a single 11G NVIDIA 2080Ti GPU. Each experiment is repeated 10 times, and the average results are reported. The total number of

Dataset	FB15k-237	WN18RR	UMLS
Entities	14541	40943	135
Relations	237	11	46
Train Edges	272115	86835	5216
Dev Edges	17535	3034	652
Test Edges	20466	652	661

Table 3: The statistics of the three benchmark datasets.

parameters of ReadE is 11.1M. It takes about 8 and 4 hours to get the best result running on FB15k-237 and WN18RR datasets, respectively.

C KG Data Augmentations for Creating Positive Pairs

In CL, a popular way to construct the positive pair on graph-structure data is to corrupt the graph structure to change the adjacency information of each entity, therefore defining the different views of the same node as the positive pair. Inspired by GraphCL (You et al., 2020), we design two types of knowledge-graph-level data augmentations to realize the corruption.

Entity Dropping. Given the knowledge graph G_e, edge dropping will randomly discard certain portion of entities (*i.e.*, nodes) and all the edges associated with them. Specifically, the probability of an entity to be chosen is defined as:

$$p_c(e_i) \propto \frac{1}{d(i)^{\frac{3}{4}}}, \tag{16}$$

where $d(i)$ is the degree of the entity e_i. The reason for using the reciprocal is that removing nodes with higher degree will impact more on the graph structure.

Relation Dropping. Relation dropping will first randomly choose a certain ratio of non-repetitive relations and remove all the edges that are included in these chosen relations. The definition of probability that a relation r to be chosen is similar to (16) with replacing the degree with the number of the edges that associated with r.

For each iteration, the random augmentations are operated twice and two different views of an entity e_i will be generated. Also,we repeatedly random sample entities and relations without replacement to make sure that a certain ratio (named ad β) of entities and relations are dropped.