Alleviating the Sparsity of Open Knowledge Graphs with Pretrained Contrastive Learning

Anonymous ACL submission

Abstract

Due to the sparsity of formal knowledge and the roughness of non-ontological construction methods, relevant facts are often missing in Open Knowledge Graphs (OpenKGs). Although existing completion methods have achieved promising performance, they do not alleviate the sparsity problem of OpenKGs. Owing to fewer training chances caused by sparse links, many few-shot and zero-shot entities cannot fully learn high-dimensional features. In this paper, we propose a new OpenKG Contrastive Learning (OKGCL) model to alleviate the sparsity with contrastive entities and relations. OKGCL designs (a) negative entities to discriminate different entities with the same relation, (b) negative relations to discriminate different relations with the same entity-pair, and (c) *self* positive samples to give zero-shot and few-shot entities chances to learn discriminative representations. Extensive experiments on benchmark datasets show the superiority of OKGCL over state-of-the-art models.

1 Introduction

Open Knowledge Graphs (OpenKGs) represent objective facts with triples in the form of ("subject noun phrase", "relation phrase", "object noun phrase"). Taking noun phrases as entities and relation phrases as relations, OpenKGs form structured knowledge that can visually express potential connections of facts. OpenKGs are extracted from text corpora with Open Information Extraction (OpenIE) tools (Fader et al., 2011; Gashteovski et al., 2019), and generally do not rely on the specification of ontology or relational schema. Although this approach has the advantage that it can be easily bootstrapped to new domains, because of the sparsity of formal grammatical knowledge and the roughness of non-ontological construction methods, relevant facts are often missing from such OpenKGs, which makes them difficult to be directly usable for end tasks like question answering (Chandrahas and Talukdar, 2021). The task of OpenKG completion aims at finding out missing relations, which has become an indispensable step in the application of OpenKGs to downstream tasks (Gupta et al., 2019; Broscheit et al., 2020).

With the great success of deep learning, many completion methods have devoted to learning potential implicit features of entities and relations. These approaches project the entities and relations into embeddings, and then predict the missing relations by calculating the similarity scores of entity-pairs. Some general completion models focus on mining structural features with linear (Bordes et al., 2013), bilinear (Wang et al., 2014; Lin et al., 2015), complex (Yang et al., 2015; Trouillon et al., 2016) or convolutional (Dettmers et al., 2018; Nguyen et al., 2018) operations, while the OpenKG-specific completion models enhance the representations with external information (Gupta et al., 2019) and pretrained language models (Chandrahas and Talukdar, 2021).

Although existing models have achieved better performance, they do not effectively tackle the sparsity problem of OpenKGs. The sparsity is mainly reflected in the imbalance of entity degrees, that is, many entities have few or zero related links in an OpenKG. According to our statistics, the degree of 54.6% entities in ReVerb20K and 89% entities in ReVerb45K is less than 3. These entities are denoted as few-shot or zero-shot entities. Due to fewer training chances caused by sparse links, few-shot or zero-shot entities are not well trained, resulting in poor generalization performance. This motivates a strong need to develop a more effective method to alleviate the sparsity of OpenKGs.

Being popular in unsupervised representation learning, *contrastive learning* aims to learn effective and discriminative representations by introducing a large number of negative samples in contrast with positive samples (He et al., 2020; Gao et al., 2021; Zhu et al., 2021). These negative samples can

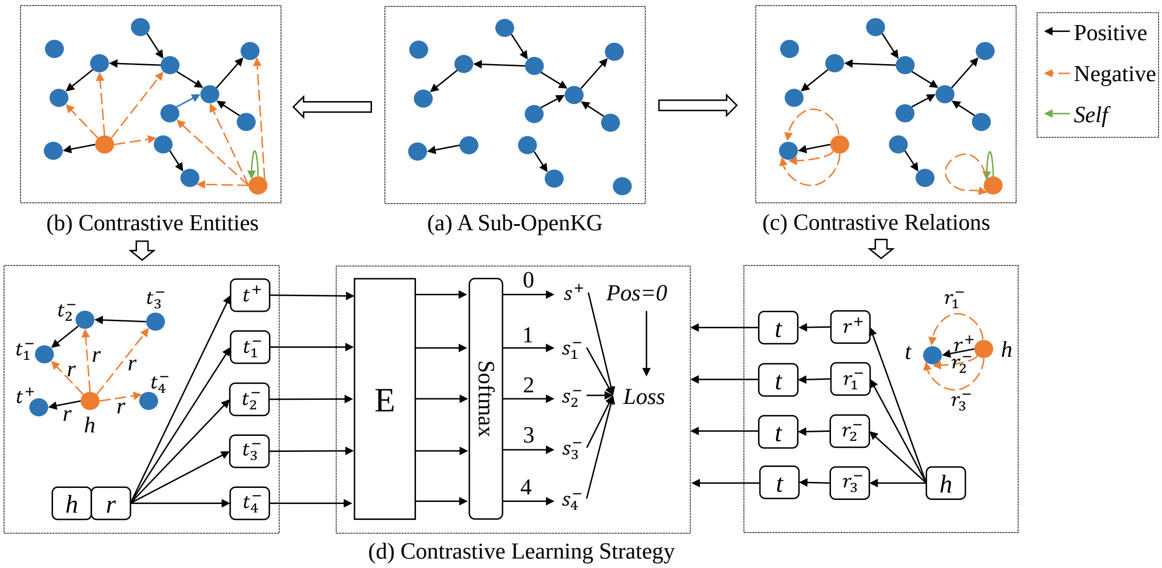


Figure 1: The proposed OKGCL model to alleviate the sparsity of OpenKGs.

enrich the understanding of positive samples in the form of negative feedback. Because sparse links make few-shot entities of OpenKGs unable to fully learn high-dimensional features, we propose to generate negative samples to contrast with the existing links, so as to learn discriminative representations of few-shot entities. With this motivation, in this work we attempt to design and incorporate negative samples to alleviate the sparsity of OpenKGs.

We propose OpenKG Contrastive Learning (OKGCL) to alleviate the sparsity of OpenKGs with negative samples. For an OpenKG (Fig. 1a), OKGCL investigates three key ideas: (1) *Contrastive Entity* to generate negative entity samples to mine discriminative features of different entities with the same relation (Fig. 1b), (2) *Contrastive Relation* to generate negative relation samples to capture discriminative features of different relations with the same entity-pair (Fig. 1c), and (3) *Contrastive Self* to construct the positive sample which gives few-shot and zero-shot entities chances to learn discriminative representations (Fig. 1b,c). Extensive experiments on benchmarks prove the superiority of OKGCL over state-of-the-art baselines. In summary, we highlight our key contributions:

- We improve the completion performance from the perspective of alleviating the sparsity problem. To our knowledge, this is the first work to alleviate the sparsity of OpenKGs without using an external source.
- We propose OKGCL, a new OpenKG contrastive learning model which generates contrastive entities and contrastive relations to alleviate the sparsity of OpenKGs. OKGCL also generates *self* positive sample to give zero-shot or few-shot entities one or more chances to be contrasted with negative samples.
- Extensive experiments show the superiority of OKGCL over the state-of-the-art baselines. We also demonstrate that OKGCL outperforms the baselines with pretrained language models on different sparsity granularity. Source code will be public later.

2 Related Work

2.1 Open Knowledge Graph

OpenKGs represent factual knowledge in structured forms, which are extracted with Open Information Extraction (OpenIE) tools (Fader et al., 2011; Gashteovski et al., 2019). They do not require the specification of ontology or relational schema, and thus can easily bootstrap to new domains. However, this rough construction makes OpenKGs sparse with many valid relations missing (Chandrahas and Talukdar, 2021). Finding out missing relations to complete the OpenKGs has thus become an important research topic.

Many completion models have been devoted to learning implicit embeddings of entities and relations. Some translation-based embedding models, such as TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015) and TransD (Ji et al., 2015) apply simple linear or bilinear operations to model the embeddings of entities and relations. DistMult (Yang et al., 2015) and ComplEx

(Trouillon et al., 2016) design similarity scoring functions to learn semantic information. ConvE (Dettmers et al., 2018) and ConvKB (Nguyen et al., 2018) apply convolutional neural network technology to learn non-linear features.

Previous OpenKG-specific completion models fake similar entities with side information and pretrained language models. CaRe (Gupta et al., 2019) tries to learn canonicalization infused embeddings, which fake similar entities of OpenKGs by integrating canonicalization and side information in an error-conscious manner. OKGIT (Chandrahas and Talukdar, 2021) employs the output of a pretrained language model to improve the CaRe model from type compatibility. However, these approaches are still limited in alleviating the sparsity of OpenKGs as shown in our experiments (§5). In contrast, we propose a contrastive learning method that is more effective and does not rely on external sources.

2.2 Contrastive Learning

Contrastive representation learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Hadsell et al., 2006; Gao et al., 2021). The effectiveness of contrastive learning is closely related to the distribution of positive and negative samples. Some methods pay attention to the choice of positive samples (Hénaff, 2020; Hjelm et al., 2019), while others devote to the generation of negative samples (Bachman et al., 2019; Ye et al., 2019; Chen et al., 2020).

Contrastive learning has achieved great success in visual representation learning (He et al., 2020), natural language processing (Gao et al., 2021) and graph representation learning (Zhu et al., 2021). A number of unsupervised graph representation learning methods attempt to leverage a contrastive learning loss at node (Velickovic et al., 2019), graph (Sun et al., 2020) and multi-view levels (Hassani and Ahmadi, 2020; Zhu et al., 2021). However, to our knowledge, none has studied contrastive learning to alleviate the sparsity problem of OpenKGs.

3 Preliminaries

• **Open Knowledge Graph** In an OpenKG $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, let a triple be (h, r, t), where $h, t \in \mathcal{E}$ represent the head and tail entities, and $r \in \mathcal{R}$ represents the relation between entities h and t; $|\mathcal{E}|$ and $|\mathcal{R}|$ are the number of entities and relations, respectively. The entities h, t and relation r are represented by non-empty word sequences; let $w_h = \{w_{h,1}, ..., w_{h,|w_h|}\}$ be the word sequence of entity h, and $w_r = \{w_{r,1}, ..., w_{r,|w_r|}\}$ be the word sequence of relation r. The representations of entities and relations are denoted as $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times D}$ and $\mathbf{R} \in \mathbb{R}^{|\mathcal{R}| \times D}$, where D is the feature dimension.

• **Relation Prediction Task in OpenKGs** The relation prediction problem of OpenKGs is to predict answer entities for two questions: (1) predicting the tail $Q_t = (h, r, ?)$ and (2) predicting the head $Q_h = (?, r, t)$. For each question, the number of possible correct answer entities is greater than or equal to one, because there could be multiple entities with the same meaning but different forms in an OpenKG (Broscheit et al., 2020). For example, for the question ("NBC-TV", "has office in", ?), we expect all answers from the set of entities {"New York", "NYC", "New York City"}.

• **Contrastive Learning Method** Contrastive learning learns high-dimensional feature representations by contrasting a positive sample with N negative samples. Specifically, for an input h, it assumes a positive pair $\mathcal{P} = (h, h^+)$, and N negative pairs $\mathcal{N} = \{(h, h_j^-)\}_{j=1}^N$. It defines the following contrastive score with the goal to push the positive pair closer in the representation space while pushing apart the representation of the negative pairs:

$$S(h) = \frac{\exp(\beta(h, h^+)/\tau)}{\sum_{n \in \{\mathcal{P}, \mathcal{N}\}} \exp(\beta(n)/\tau)} \quad (1)$$

where $\beta(h_1, h_2)$ is a similarity function and τ is a temperature hyperparameter.

4 Proposed OKGCL Model

In this section, we present our contrastive learning model, **OKGCL**. The overall framework of the OKGCL is shown in Fig. 1. In §4.1, we introduce a simple but effective embedding model to fuse the textual and structural features of OpenKGs. In §4.2, we design contrastive entities and relations to optimize the above embedding model. Finally, the training procedure is given in §4.3.

4.1 Embedding Model

For a triple $(h, r, t) \in \mathcal{G}$, the initial representations of entities h, t and relation r is defined as $e_h, e_t \in \mathbf{E}$ and $e_r \in \mathbf{R}$, respectively. For the pair (h, r), the word sequence of head entity h is $w_h = \{w_{h,1}, ..., w_{h,|w_h|}\}$, and that of relation r is $w_r = \{w_{r,1}, ..., w_{r,|w_r|}\}$. We encode each of these sequences with a single layer bi-directional

Gated Recurrent Unit (BiGRU) (Cho et al., 2014).

$$e_i^w = \text{BiGRU}(w_i) \text{ for } w_i \in \{w_h, w_r\} \quad (2)$$

The embedding (i.e., hidden state) of the first token $w_{i,1}$ is taken as the final embedding of the sequence w_i, that is $e_i^w = e_i^w[1]$. This way we get the textual embeddings of head entity h and relation r as e_h^w and e_r^w, respectively.

Then, we focus on exploiting potential connections between entities and relations. We use a two-dimensional convolutional network (Dettmers et al., 2018) to learn the potential connections between a head entity h and a relation r as follows:

$$\varphi(h,r) = \sigma(\text{Linear}(\sigma([\widehat{e_h};\widehat{e_r}] * \omega))) \quad (3)$$

where $*$ denotes a two-dimensional convolutional layer with filters ω, Linear projects the dimension to D, and σ represents a ReLU activation; $\widehat{e_h}, \widehat{e_r} \in \mathbb{R}^{D_1 \times D_2}$ are reshaped from $[e_h + e_h^w; e_r^w] \in \mathbb{R}^{2D}$ with $2D = D_1 D_2$; Through the convolution module, the potential embeddings of entity h and relation r are jointly encapsulated.

Finally, we compute the similarity score for each triple (h,r,t). The similarity score is computed with a cosine similarity function as:

$$\beta(h,r,t) = \frac{\varphi(h,r)^\top e_t}{||\varphi(h,r)|| \cdot ||e_t||} \quad (4)$$

where $e_t \in \mathbf{E}$ is initial embedding of tail entity t.

4.2 Contrastive Learning Model

Taking the model in §4.1 as a basic embedding module, we design OKGCL, a contrastive learning model to alleviate the sparsity of OpenKGs (Fig. 1d). It has four modules: (1) *Contrastive Entity* contrasts negative entities with positive entities, to mine features of different entities with the same relation. (2) *Contrastive Relation* contrasts negative relations with positive relations, to capture potential features of different relations with the same head and tail entities. (3) *Contrastive Fusion* fuses negative entities and negative relations to contrast with positive samples. (4) *Contrastive Self* constructs the positive sample $(h, \textit{self}, h^+)$ and contrasts this positive sample with negative samples, to give zero-shot or few-shot entities chances to learn discriminative representations through training.

Contrastive Entity We alleviate the sparsity of OpenKGs from the perspective of *nagative* entity. The triples in the OpenKG are regarded as positive samples, while negative samples are generated with contrastive entities. For a positive sample $\mathcal{P}_e = (h, r, t^+)$, a contrastive entity t_j^- is randomly selected from the entity list $\mathcal{E} - \mathcal{E}(h,r)$, where $\mathcal{E}(h,r)$ is the entity list of true answers, that is, the triple $(h,r,t_i) \in \mathcal{G}$ if entity $t_i \in \mathcal{E}(h,r)$. Fig. 1b gives two examples of contrastive entities. For each positive sample $\mathcal{P}_e = (h,r,t^+)$, its negative samples are generated with multiple contrastive entities:

$$\mathcal{N}_e = \{(h,r,t_j^-)\}_{j=1}^{N_e} \quad (5)$$

where N_e is the number of contrastive entities. The contrastive score for the positive entity $\mathcal{P}_e$ is:

$$S(h,r,t^+) = \frac{\exp(\beta(h,r,t^+)/\tau)}{\sum_{n \in \{\mathcal{P}_e, \mathcal{N}_e\}} \exp(\beta(n)/\tau)} \quad (6)$$

where $\beta(.)$ is the similarity score as in Eq. (4).

Contrastive Relation We also alleviate the sparsity of OpenKGs from the perspective of *negative* relation. The triples in the OpenKG are regarded as positive samples, while negative samples are generated with contrastive relations. For a positive sample $\mathcal{P}_r = (h, r^+, t)$, a contrastive relation r_j^- is randomly selected from the relation list $\mathcal{R} - \mathcal{R}(h,t)$, where $\mathcal{R}(h,t)$ is a relation list that satisfies the condition: triple $(h, r_i, t) \in \mathcal{G}$ if relation $r_i \in \mathcal{R}(h,t)$. Fig. 1c gives two examples of contrastive relations. For each positive sample $\mathcal{P}_r = (h, r^+, t)$, its negative samples are generated with multiple contrastive relations:

$$\mathcal{N}_r = \{(h, r_j^-, t)\}_{j=1}^{N_r} \quad (7)$$

where N_r is the number of contrastive relations. The contrastive score for the positive relation $\mathcal{P}_r$ is:

$$S(h,r^+,t) = \frac{\exp(\beta(h,r^+,t)/\tau)}{\sum_{n \in \{\mathcal{P}_r, \mathcal{N}_r\}} \exp(\beta(n)/\tau)} \quad (8)$$

Contrastive Fusion In this module, the negative entity samples and negative relation samples, which can be generated from the Contrastive Entity and Contrastive Relation modules, are fused to alleviate the sparsity of OpenKGs. The contrastive score for a positive sample $\mathcal{P}_f = (h, r^+, t^+)$ is:

$$S(h,r^+,t^+) = \frac{\exp(\beta(h,r^+,t^+)/\tau)}{\sum_{n \in \{\mathcal{P}_f, \mathcal{N}_e, \mathcal{N}_r\}} \exp(\beta(n)/\tau)} \quad (9)$$

where the positive sample $\mathcal{P}_f = (h, r^+, t^+)$ is a triple in the OpenKG, the negative samples are achieved by merging the negative entity samples $\mathcal{N}_e$ and negative relation samples $\mathcal{N}_r$.

Contrastive Self Due to sparsity, there are many entities in an OpenKGs with only few or zero links, referred to as few- or zero-shot entities. Compared with those entities with many links, few-shot entities are often not fully trained because of fewer links, and zero-shot entities cannot be trained because they have no links with rest of the OpenKG.

In view of the above problem, we propose to add the *Self* relation to construct the positive sample $\mathcal{P}_s = (h, \textit{self}, h^+)$. The embeddings of the same entity h and h^+ are different, where the embedding of h^+ is the initial embedding e_h, and that of h is the sum of textual embedding e_h^w and initial embedding e_h. For such a positive sample $\mathcal{P}_s = (h, \textit{self}, h^+)$, negative entity samples $\mathcal{N}_e = \{(h, \textit{self}, h_j^-)\}_{j=1}^{N_e}$ can be generated with contrastive entities, which are randomly selected from the entity list $\mathcal{E} - h$. Similarly, negative relation samples $\mathcal{N}_r = \{(h, \textit{self}_j^-, h^+)\}_{j=1}^{N_r}$ can be generated with contrastive relations, which are randomly selected from the relation list $\mathcal{R} - \textit{self}$. The contrastive score for a positive sample $\mathcal{P}_s = (h, \textit{self}, h^+)$ can be computed with Eq. (9). The examples of contrastive self are shown in Fig. 1b-c. Through the *Self* positive sample, these zero-shot or few-shot entities can have one or more chances to be trained with contrastive learning.

4.3 Training Procedure

The proposed OKGCL model is trained in Pretrain and Finetune stages, where the Pretrain stage aims to learn discriminative representation with contrastive entities and contrastive relations, and the Finetune stage aims to select optimal hyperparameters for specific end tasks.

Pretrain The training objective is as follows:

$$L_{\mathcal{P}} = -\frac{1}{|\mathcal{P}|}\sum_{i=1}^{|\mathcal{P}|}\log(S(i))$$
$$\text{where } S(i) = \begin{cases} S(h, r, t^+), \text{ if Entity} \\ S(h, r^+, t), \text{ if Relation} \\ S(h, r^+, t^+), \text{ if Fusion} \end{cases} \tag{10}$$

where $|\mathcal{P}|$ is the number of all positive samples.

Finetune The pretrained model is finetuned to adapt for the relation prediction task of OpenKGs. Note that, for a pair (h, r), there could be more than one true object entities in an OpenKG, where the object entities are put into $\mathcal{E}(h, r)$. For a test triple (h_i, r_i, t_i), the representation $\varphi(h_i, r_i)$ of entity h_i and relation r_i is matched with the embeddings $\mathbf{E}$ of all entities via a matrix multiplication as.

$$X_i = \delta(\varphi(h_i, r_i)\mathbf{E}^\top) \tag{11}$$

where δ represents a Sigmoid activation. We use a binary cross-entropy loss to optimize the parameters. The loss function can be defined as:

$$\begin{aligned} L_{\mathcal{F}} = -\frac{1}{|\mathcal{M}||\mathcal{E}|}\sum_{i=1}^{|\mathcal{M}|}\sum_{j=1}^{|\mathcal{E}|}(Y_{i,j} \cdot \log X_{i,j} \\ +(1 - Y_{i,j}) \cdot \log(1 - X_{i,j})) \end{aligned} \tag{12}$$

$$\text{with } \begin{cases} Y_{i,j} = 1, \text{ if } (h_i, r_i, t_j) \in \mathcal{E}(h_i, r_i) \\ Y_{i,j} = 0, \text{ if } (h_i, r_i, t_j) \notin \mathcal{E}(h_i, r_i) \end{cases}$$

where $|\mathcal{E}|$ is the number of all entities, $|\mathcal{M}|$ is the number of all test samples.

5 Experiments

5.1 Datasets

The statistics of datasets are summarized in Table 1. ReVerb20K and ReVerb45K are OpenKG benchmark datasets (Vashishth et al., 2018), which are constructed through the ReVerb open knowledge base (Fader et al., 2011). The ReVerb45K with 27.0K entities and 21.6K relations, is larger and sparser than the ReVerb20K with 11.1K entities and 11.1K relations. In the sparsity problem handling, we conduct variants of training sets at different sparsity granularity ([100%, 80%, 60%, 40%, 20%]), where *x*% represents to randomly remove the percentages (1-*x*%) of samples from the original training sets. Validation sets and test sets are the same for different sparsity granularity. As can be seen from Table 1, the average degree decreases with the decrease of sparsity granularity.

5.2 Evaluation Protocols

For single test triple, we use the classic Mention Ranking (Gupta et al., 2019; Broscheit et al., 2020) as evaluation protocol, which is the minimum ranking position of all answer entities. For all test triples, we use three most commonly used ways to integrate the above Mention Ranking scores. *H@N*: The proportion of answer entity ranking in the top N position. *AR*: Average all ranking scores. *ARR*: Compute the reciprocal of each ranking score, and average all the reciprocals. A model with better performance should have higher $H@N$, higher ARR and lower AR.

Dataset	Entity	Relation	Cluster	Valid	Test	Train					Average Degree				
						100%	80%	60%	40%	20%	100%	80%	60%	40%	20%
ReVerb20K	11.1K	11.1K	10.8K	1.6K	2.4K	15.5K	12.4K	9.3K	6.2K	3.1K	1.4	1.1	0.8	0.6	0.3
ReVerb45K	27.0K	21.6K	18.6K	3.6K	5.4K	36.0K	28.8K	21.6K	14.4K	7.2K	1.3	1.1	0.8	0.5	0.3

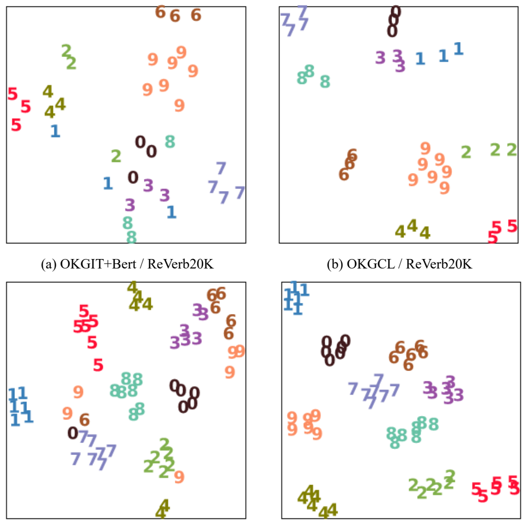
Table 1: The statistics of Datasets.

Type	Model	ReVerb20K						ReVerb45K					
		$AR\downarrow$	ARR	$H@1$	$H@10$	$H@50$	$H@100$	$AR\downarrow$	ARR	$H@1$	$H@10$	$H@50$	$H@100$
General	TransE	1497	13.3	2.2	29.6	43.0	49.2	2222	15.8	9.3	25.9	37.1	43.2
	DistMult	4569	1.9	1.3	2.7	5.2	7.1	5782	8.5	7.7	9.7	12.0	13.6
	ComlEx	4376	2.0	1.4	3.0	5.6	7.7	5173	8.9	7.5	11.3	16.0	18.9
	ConvE	1085	25.5	19.9	35.8	50.1	57.2	2483	22.1	16.6	32.4	43.3	47.9
	ConvTransE	1080	26.1	20.5	35.9	50.0	57.1	2490	23.4	17.9	33.8	44.4	48.8
OpenKG	CaReTransE	950	30.3	23.2	42.8	58.4	64.6	2414	19.5	7.8	37.5	47.5	51.4
	CaReConvE	801	31.6	25.6	42.9	56.7	63.4	1589	29.7	23.4	41.3	53.6	58.7
OpenKG+ Pretrained	OKGIT+Bert	516	34.8	27.4	48.0	64.4	71.2	813	32.2	**25.7**	44.6	58.1	63.9
	OKGIT+Rob	545	34.7	27.2	48.1	65.5	72.0	864	32.0	25.3	44.7	57.3	62.6
Our	OKGCL	**421**	**38.1**	**29.9**	**53.3**	**68.5**	**75.2**	**744**	**33.2**	25.6	**48.1**	**61.9**	**67.7**

Table 2: The results on ReVerb20K and ReVerb45K test data.

5.3 Models and Settings

To prove the effectiveness of the proposed model, we compare several high-performing General models: **TransE** (Bordes et al., 2013), **DistMult** (Yang et al., 2015), **ComplEx** (Trouillon et al., 2016), **ConvE** (Dettmers et al., 2018), **ConvTransE** (Shang et al., 2019), and OpenKG-specific models: **CaReTransE** (Gupta et al., 2019), **CaReConvE** (Gupta et al., 2019), **OKGIT+Bert** and **OKGIT+Rob** (Chandrahas and Talukdar, 2021). The results of baselines are reproduced with open source implementations. **OKGCL** is our proposed model, where the optimizer is set to Adam, the embedding size is set to 300. The entity and relation embeddings are initialized randomly, and the word vectors are initialized with the GloVe embeddings.



(a) OKGIT+Bert / ReVerb20K (b) OKGCL / ReVerb20K (c) OKGIT+Bert / ReVerb45K (d) OKGCL / ReVerb45K

Figure 2: The visualization results, where the same numbers represent entities with the same meaning.

5.4 Results

Overall Performance We evaluate the overall performance of the proposed OKGCL. The results on ReVerb20K and ReVerb45K datasets are given in Table 2, where the best score is in bold. OKGCL achieves substantial improvements in comparison to baselines. For ReVerb20K, OKGCL outperforms all baselines with strong improvements on all metrics. Among that, ARR metric increases by 3.3 point and H@1,10,50,100 metrics increase by 2.5, 5.2, 3.0, 3.2 points. For ReVerb45K, the performance of OKGCL is better than that of the General and OpenKG baselines with strong improvements on all metrics. Among that, ARR metric increases by 3.5 point and H@1,10,50,100 metrics increase by 2.2, 6.8, 8.3, 9.0 points. Note that OKGCL outperforms the OpenKG+Pretrained baselines on all metrics for ReVerb20K and most metrics for ReVerb45K. OKGCL with simple structures and contrastive learning strategies, can achieve better performance at lower costs, than the OpenKG+Pretrained baselines with complex structures and pretrained language models, which is a very favorable discovery. In summary, OKGCL effectively capture potential discriminative features of OpenKGs with contrastive entities and relations.

Performance w.r.t. Visualization We use the t-SNE visualization to prove that the proposed

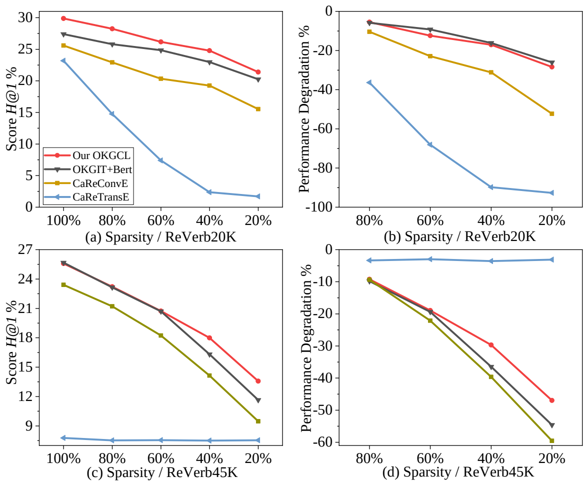


Figure 3: The results of sparsity granularity.

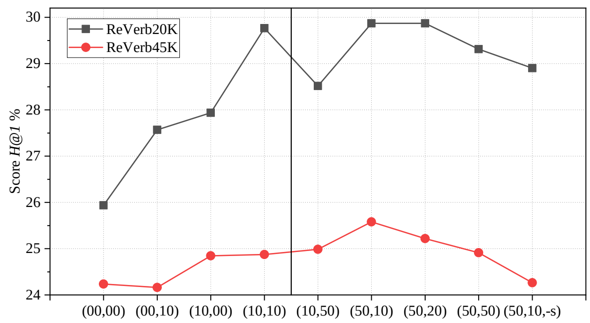


Figure 4: The results of the number collocations of contrastive entities and contrastive relations.

OKGCL can make the entities with the same meaning closer in spatial distribution. For this visualization experiment, we selected 10 entity clusters for each dataset, where each cluster has more than three entities. The state-of-the-art model, OKGIT+Bert, is as the baseline to compare with OKGCL. Fig. 2 shows the visualization results on ReVerb20K and ReVerb45K datasets, where Fig. 2a and Fig. 2c are for OKGIT+Bert and Fig. 2b and Fig. 2d are for OKGCL. Through observing the distributions of entities in Fig. 2, we found that the embeddings of entities with the same meaning (same number) are closer in OKGCL than in OKGIT+Bert. The qualitative results are consistent with the quantitative results in Table 2, which further verify the effectiveness of OKGCL.

Performance w.r.t. Sparsity The motivation in this paper is to alleviate the sparsity problem of OpenKGs, so we hope OKGCL can perform well on sparse data. We conduct experiments of different sparsity granularity ([100%, 80%, 60%, 40%, 20%]). Fig. 3 gives the results of different sparsity granularity on ReVerb20K and ReVerb45K datasets, where Fig. 3a and Fig. 3c show $H@1$ scores, Fig. 3b and Fig. 3d show the percentage of performance degradation ((x%-100%)/100%).

First, we analyze the $H@1$ scores of different models on different sparsity granularity. OKGCL achieves the best scores in all sparsity granularity than CaReConvE and CaReTransE on both datasets. For ReVerb20K (Fig. 3a), OKGCL achieves the $H@1$ scores of 28.2, 26.2, 24.8, 21.4 in the sparsity granularity of 80%, 60%, 40%, 20%, respectively, which gains improvements over OKGIT+Bert at 2.4, 1.3, 1.8 and 1.1 points, respectively. For ReVerb45K (Fig. 3c), the $H@1$ scores of OKGCL and OKGIT+Bert are similar in the sparsity granularity of 100%, 80%, 60%. When the sparse granularity decreases to 40% and 20%, the $H@1$ scores of OKGCL becomes better than that of OKGIT+Bert. This shows that OKGCL performs better than all baselines on sparse data, even OKGIT+Bert enhanced with pretrained language models.

Second, we analyze the percentage of performance degradation. For ReVerb45K (Fig. 3d), the performance degradation of OKGCL is better than OKGIT+Bert and CaReConvE models. CaReTransE seems to have the best performance degradation, but there is no room to step back due to the low $H@1$ score. For ReVerb20K (Fig. 3b), the performance degradation of OKGCL is better than that of CaReTransE and CaReConvE, but is almost similar to that of OKGIT+Bert. Note that OKGIT+Bert enhances the representation learning with pretrained language models, which have loaded a lot of commonsense knowledge with the training of large-scale data and a great quantity of hyperparameters. OKGCL does not introduce side information and pretrained language models, but its performance degradation is similar or better than that of OKGIT+Bert. The above experiments and analysis demonstrate the effectiveness of OKGCL in alleviating the sparsity problem of OpenKGs.

Performance w.r.t. Number of Contrastive Entities and Relations The Fusion module in §4.2 pays attention to fusing negative samples of contrastive entities and relations. Our assumption is that the number of contrastive entities and relations could affect the performance of OKGCL. So, we design several collocations of contrastive entities (left) and contrastive relations (right) with different numbers: (00, 00), (00, 10), (10, 00), (10, 10), (10, 50), (50, 10), (50, 20), (50, 50).

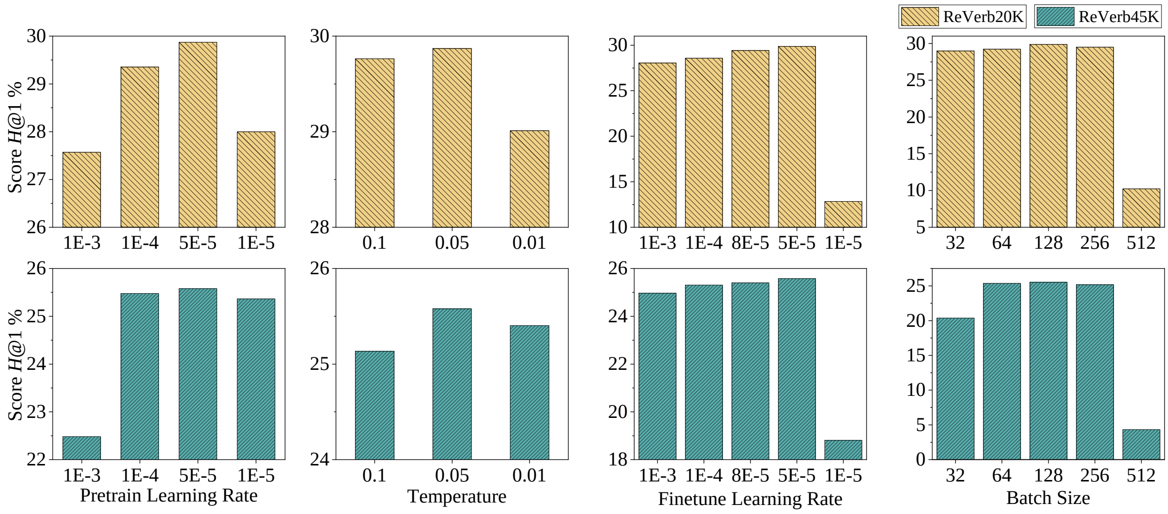


Figure 5: The results of hyperparameters.

Fig. 4 gives the results of number collocations on ReVerb20K and ReVerb45K datasets. Through observation, we have several findings: First, contrastive entities are more important than contrastive relations in improving model performance. For example, the performance of (10, 00) is better than that of (00, 10), and the performance of (50, 10) is also better than that of (10, 50). Second, the role of contrastive relations can be more stimulated with the help of contrastive entities. For example on ReVerb45K, the performance of (00, 10) is worse than that of (00, 00) without any contrastive entities and relations, while the performance of (10, 10) is a little better than that of (10, 00). Third, OKGCL achieves the best result with the number collocation of (50,10) on both datasets. The above collocations also give the results of ablation study, where (00, 00) removes all contrastive modules, (00, 10) removes Contrastive Entity module, and (10, 00) removes Contrastive Relation module. The performance decreases when any module is removed, which proves the effectiveness of each module.

Performance w.r.t. Contrastive Self The results of removing the *self* relation and its related positive and negative samples, are shown in (50,10,-s) of Fig. 4. Compared with the H@1 scores of (50,10), the performance of OKGCL decreases prominently when the Contrastive Self is removed (50,10,-s), especially for the sparser ReVerb45K. This proves the effectiveness of Contrastive Self.

Performance w.r.t. Hyperparameters We explore the sensitivity of OKGCL to hyperparameters. The list of hyperparameters are from two aspects: (1) Hyperparameters for Pretrain stage: learning rate $\in$ {1e-3, 1e-4, 5e-5, 1e-5}, temperature regulation value $\in$ {0.1, 0.05, 0.01}. (2) Hyperparameters for Finetune stage: learning rate $\in$ {1e-3, 1e-4, 8e-5, 5e-5, 1e-5}, batch size $\in$ {32, 64, 128, 256, 512}. The results of hyperparameters are shown in Fig. 5. For Pretrain stage, the optimal value of learning rate is 5e-5 on both datasets, where too large a learning rate, such as 1E-3, may skip the optimal results. And the optimal value of temperature regulation value is 0.05 on both datasets. For Finetune stage, the optimal value of learning rate is 5e-5 on both datasets, where too small a learning rate, such as 1E-5, brings worse performance than too large a learning rate, such as 1E-3. The optimal value of batch size is 128 on both datasets, where OKGCL is not very sensitive to batch size, but too large a batch size could have a negative impact.

6 Conclusion

In this paper, we provide empirical insights about the sparsity of OpenKGs, and propose a new contrastive learning model, OKGCL, to alleviate the sparsity of OpenKGs with contrastive entities and contrastive relations. Through extensive experiments and comprehensive analysis on real-world datasets, the proposed OKGCL achieves better performance than state-of-the-art models.

According to observation, there may be more than one object entity for a pair (h, r) in an OpenKG, which produces multiple positive samples. That challenges the existing contrastive learning mechanism with only one positive sample. As part of future work, we hope to improve the contrastive learning to deal with the above problem.

References

Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15509–15519.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. 2020. Can we predict new facts with open knowledge graph embeddings? A benchmark for open link prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL, Online, July 5-10, 2020*, pages 2296–2308.

Chandrahas and Partha P. Talukdar. 2021. OKGIT: open knowledge graph link prediction with implicit types. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP, Online Event, August 1-6*, pages 2546–2559. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1535–1545.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *CoRR*, abs/2104.08821.

Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. 2019. OPIEC: an open information extraction corpus. In *1st Conference on Automated Knowledge Base Construction, AKBC, Amherst, MA, USA, May 20-22, 2019*.

Swapnil Gupta, Sreyash Kenkre, and Partha P. Talukdar. 2019. Care: Open knowledge graph embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Hong Kong, China, November 3-7, 2019*, pages 378–388.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1735–1742. IEEE Computer Society.

Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive multi-view representation learning on graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4116–4126. PMLR.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE.

Olivier J. Hénaff. 2020. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4182–4192. PMLR.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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 of the Asian Federation of Natural Language Processing, ACL, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 687–696.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2181–2187.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. 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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 327–333.

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 *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3060–3067.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2020. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080.

Shikhar Vashishth, Prince Jain, and Partha P. Talukdar. 2018. CESI: canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW, Lyon, France, April 23-27, 2018*, pages 1317–1327.

Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep graph infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27-31, 2014, Québec City, Québec, Canada*, pages 1112–1119.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6210–6219. Computer Vision Foundation / IEEE.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2069–2080. ACM / IW3C2.