CAKE: A Scalable Commonsense-Aware Framework For Multi-View Knowledge Graph Completion

Guanglin Niu[1,2], **Bo Li**[2,3], **Yongfei Zhang**[1,4]*, **Shiliang Pu**[5]

[1] Beijing Key Laboratory of Digital Media, Beihang University, Beijing, China
[2] Institute of Artificial Intelligence, Beihang University, Beijing, China
[3] Hangzhou Innovation Institute, Beihang University, Hangzhou, China
[4] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China [5] Hikvision Research Institute, Hangzhou, China
{beihangngl, boli, yfzhang}@buaa.edu.cn, pushiliang.hri@hikvision.com

Abstract

Knowledge graphs store a large number of factual triples while they are still incomplete, inevitably. The previous knowledge graph completion (KGC) models predict missing links between entities merely relying on fact-view data, ignoring the valuable commonsense knowledge. The previous knowledge graph embedding (KGE) techniques suffer from invalid negative sampling and the uncertainty of fact-view link prediction, limiting KGC's performance. To address the above challenges, we propose a novel and scalable **Commonsense-Aware Knowledge Embedding** (**CAKE**) framework to automatically extract commonsense from factual triples with entity concepts. The generated commonsense augments effective self-supervision to facilitate both high-quality negative sampling (NS) and joint commonsense and fact-view link prediction. Experimental results[1] on the KGC task demonstrate that assembling our framework could enhance the performance of the original KGE models, and the proposed commonsense-aware NS module is superior to other NS techniques. Besides, our proposed framework could be easily adaptive to various KGE models and explain the predicted results.

1 Introduction

In recent years, knowledge graphs (KGs) such as Freebase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2015) and NELL (Mitchell et al., 2018) have been widely used in many knowledge-intensive applications, including question answering (Sun et al., 2020; Saxena et al., 2020), dialogue systems (Yang et al., 2020; Zhou et al., 2018) and recommender systems (Wang et al., 2021, 2019a). However, the KGs constructed manually or automatically are inevitably incomplete, requiring KGC to infer new facts.

---

*Corresponding author.
[1]The source code and datasets of this paper are available at https://github.com/ngl567/CAKE.
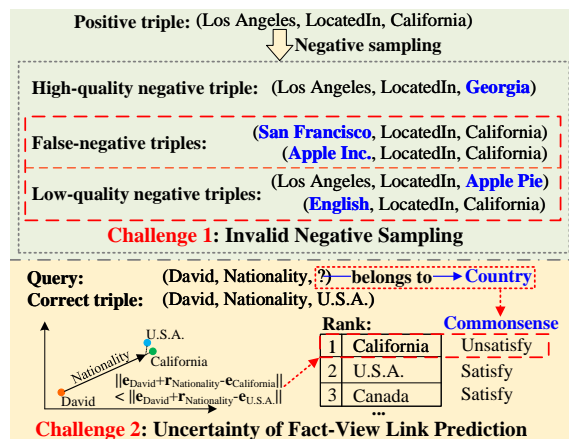

Figure 1: Two examples exhibit the challenges that needed to be addressed. **Challenge 1**: Given a positive triple, some generated negative triples are false-negative or low-quality. **Challenge 2**: For link prediction, the entity $California$ ranks higher than the correct entity $U.S.A.$ due to the uncertainty of KG embeddings. However, the correct answer entity should belong to the concept $Country$ in the view of commonsense.

The previous KGC models can be classified into three main streams: (1) Rule learning-based models mine logic rules for induction reasoning, such as AMIE+ (Galárraga et al., 2015), DRUM (Sadeghian et al., 2019) and AnyBurl (Meilicke et al., 2019). (2) Path-based models (Liu et al., 2020; Xiong et al., 2017; Lin et al., 2018) search paths for multi-hop reasoning. (3) KGE models such as TransE (Bordes et al., 2013) and its variants (Sun et al., 2019; Zhang et al., 2019a, 2020) learn the embeddings of entities and relations to score the plausibility of triples for link prediction.

Among all the existing KGC models, KGE approaches achieve higher efficiency and better performance. Specifically, the KGE-based KGC pipeline can be divided into two stages: learning knowledge graph (KG) embeddings at the training and link prediction at the inference. Learning KG embeddings relies on a basic procedure of negative sampling (Li et al., 2021). Link prediction aims to

infer the missing entity or relation in a triple via ranking the candidate triples' scores in virtue of the learned KG embeddings.

However, the two separate stages both have drawbacks: (1) **Invalid negative sampling**: all the previous NS (Wang et al., 2014; Cai and Wang, 2018; Sun et al., 2019; Zhang et al., 2019b; Denis et al., 2015) cannot avoid sampling the false-negative triples and low-quality negative triples, simultaneously. For instance, given the positive triple $(Los\ Angeles,\ LocatedIn, California)$ as shown in Figure 1, the existing NS strategies might sample the corrupted triples such as $(San\ Francisco, LocatedIn, California)$, which is actually a missing correct triple namely false-negative triple. On the other hand, the quality of some generated negative triples such as $(San\ Francisco, LocatedIn, Apple\ Pie)$ is too poor to make little sense for training the KGE models. (2) **Uncertainty of fact-view link prediction**: performing link prediction solely based on facts in a data-driven fashion suffers from uncertainty due to the deviation of KG embeddings compared to the symbolic representations, limiting the accuracy of KGC. Take the tail entity prediction $(David, Nationality, ?)$ in Figure 1 as an instance. The correct tail entity should belong to the concept $Country$ in the view of commonsense. Whereas the entity $California$ that is inconsistent with commonsense even ranks highest via scoring the candidate triples with KG embeddings.

Last but not least, although some KGE approaches exploit external information, including entity types (Xie et al., 2016b), textual descriptions (Xie et al., 2016a) and images of entities (Xie et al., 2017). Such auxiliary information is hard to access and enhances the single representation of entities rather than providing the semantics of commonsense. However, the valuable commonsense is always acquired by the expensive hand annotation (Rajani et al., 2019), so its high cost leads to relatively low coverage. Besides, the existing large-scale commonsense KGs such as Concept-Net (Speer et al., 2017) only contain the concepts without the links to the corresponding entities, causing them unavailable to the KGC task.

To address the above challenges, we propose a novel and scalable **C**ommonsense-**A**ware **K**nowledge **E**mbedding (**CAKE**) framework to improve the NS in the training of KGE and boost the performance of KGC **benefited from the self-**

supervision of commonsense. In specific, we attempt to automatically construct explicit commonsense via an instance abstraction technique from KGs. Then, contrary to random sampling, we **purposefully** generate the high-quality negative triples by taking advantage of the commonsense together with the characteristics of complex relations. Furthermore, a multi-view link prediction is conducted to determine the entity candidates that belong to the correct concepts in the commonsense view and predict the answer entities with the learned KG embeddings from the perspective of fact. In summary, the contributions of our work are three-fold:

- We propose a **scalable** KGC framework with an **automatic commonsense generation** mechanism to extract valuable commonsense from factual triples and entity concepts.

- We develop a **commonsense-aware negative sampling** strategy for generating valid and high-quality negative triples. Meanwhile, a **multi-view link prediction** mechanism is proposed to improve the accuracy of KGC.

- Extensive experiments on four benchmark datasets illustrate **the effectiveness and the scalability** of our whole framework and each module. We promise to release all the codes and datasets when this paper is published.

2 Related Work

2.1 KGC Models

The existing KGC models can be classified into three main categories: (1) Rule learning-based algorithms such as AMIE+ (Galárraga et al., 2015), DRUM (Sadeghian et al., 2019) and AnyBurl (Meilicke et al., 2019) automatically mine logic rules from KGs and apply these rules for inductive link prediction. However, these models are inefficient due to the time-consuming rule searching and evaluation. (2) Path-based models search paths linking head and tail entities, including path ranking approaches (Lao et al., 2011; Liu et al., 2020) and reinforcement learning-based models (Xiong et al., 2017; Lin et al., 2018). Whereas, multi-hop path-based models also spend much time in path searching. (3) KG embedding (KGE) models such as TransE (Bordes et al., 2013), RESCAL (Nickel et al., 2011), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019) and HAKE (Zhang et al.,

2020) learn the embeddings of entities and relations to score the plausibility of triples for predicting the missing triples efficiently. KGE approaches achieve higher efficiency and better performance on KGC compared with the others. However, the natural uncertainty of embeddings limits the precision of KGC relying solely on facts. More specifically, the KGE models generally need a primary negative sampling (NS) procedure to randomly or purposely sample some triples that are not observed in the KG as negative triples for training (Li et al., 2021).

2.2 Negative Sampling of KGE

Following the local closed-world assumption (Dong et al., 2014), the existing NS techniques for KGE can be classified into five categories: (1) Randomly and uniformly sampling: the majority of the KGE models generate negative triples via randomly replacing an entity or relation in a positive triple from a uniform distribution (Wang et al., 2014). (2) Adversarial-based sampling: KBGAN (Cai and Wang, 2018) integrates the KGE model with softmax probabilities to select the high-quality negative triples in an adversarial training framework. Self-adversarial sampling (Sun et al., 2019) performs similar to KBGAN, but it utilizes a self-scoring function without a generator. (3) Domain-based sampling: domain-based NS (Wang et al., 2019b) and type-constrained NS (Denis et al., 2015) both leverage domain or type constraints on sampling the corrupted entities that belong to the correct domain. (4) Efficient sampling: NSCaching (Zhang et al., 2019b) employs cache containing candidates of negative triples to improve the efficiency of sampling. (5) None-sampling: NS-KGE (Li et al., 2021) eliminates the NS procedure by converting loss functions of KGE into a unified square loss.

However, all the previous NS algorithms cannot address the issue of false-negative triples since these NS techniques, except for none sampling, would attempt to sample the corrupted triples with higher probability while they might be correct and just missing in the KG. Domain-based NS relies heavily on the constraint of the single type rather than the commonsense, limiting the diversity of negative triples. KBGAN introduces generative adversarial networks (GAN) in the NS framework, making the original model more complex and hard to train. None sampling has to convert each original KGE model into square loss, which weakens the performance of KGE models. These drawbacks

of the NS strategies degrade the training of KGE and further limit the performance of KGC.

2.3 Commonsense Knowledge Graph

Different from the factual triples, commonsense could inject rich abstract knowledge into KGs. However, the valuable commonsense is hard to access due to the costly hand annotation. In recent years, many researches attempt to construct general commonsense graphs such as ConceptNet (Speer et al., 2017), Microsoft Concept Graph (Ji et al., 2019) and ATOMIC (Sap et al., 2019). However, these commonsense graphs only contain the concepts without the links to the corresponding entities, causing them inapplicable to the KGC task. On the other hand, although some KGE models such as JOIE (Hao et al., 2019) employ the ontology built-in most of the KGs, i.e., NELL (Mitchell et al., 2018) and DBpedia (Lehmann et al., 2015), the relations in ontology such as isA, $partOf$ and $relatedTo$ mainly represent the type hierarchy but not the explicit commonsense. Such relations are useless for KGC because there are few overlaps between the ontological and the factual relations.

3 Methodology

In this section, we introduce our novel and scalable CAKE framework. As shown in Figure 2, the entire pipeline consists of three developed modules: the automatic commonsense generation (ACG) module, the commonsense-aware negative sampling (CANS) module and the multi-view link prediction (MVLP) module. Firstly, the ACG module extracts the commonsense from the factual triples with the entity concepts via an instance abstraction mechanism (§ 3.2). Then, the CANS module employs the generated commonsense to produce the high-quality negative triples, which takes the characteristics of complex relations into account (§ 3.3). Afterwards, our approach feeds the positive and the weighted negative triples into the KGE model for learning entity and relation embeddings (§ 3.4). Finally, the MVLP module conducts link prediction in a coarse-to-fine fashion by filtering the candidates in the view of commonsense and predicting the answer entities with KG embeddings from the candidates in the view of fact (§ 3.5).

3.1 Notations and Problem Formalization

Commonsense. Commonsense has gained widespread attraction from its successful use in
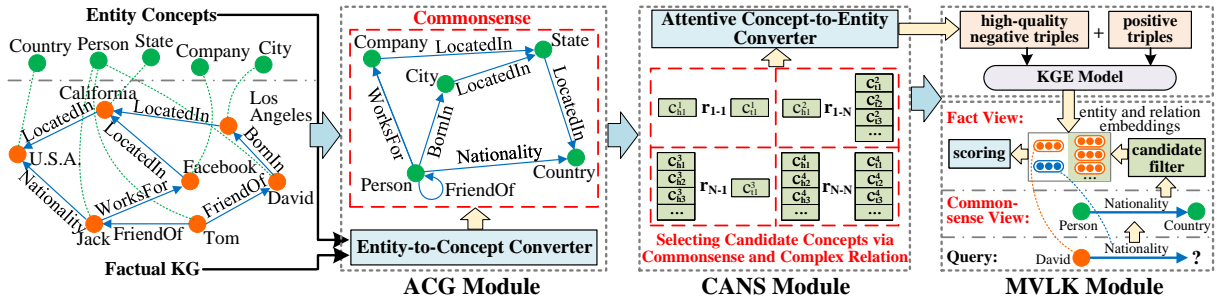
Figure 2: An overview of the CAKE framework. The orange dotes indicate the entities. The green dotes represent the entity concepts. In the CANS module, r_{1-1}, r_{1-N}, r_{N-1} and r_{N-N} denote the diverse complex relations of 1-1, 1-N, N-1 and N-N, respectively. c_{hi}^j and c_{ti}^j indicate the i-th head concept and tail concept that are selected by the commonsense and the characteristics of complex relations specific to the j-th relation.

understanding high-level semantics, which is generally represented as the concepts with their ontological relations in some well-known commonsense graphs such as ConceptNet (Speer et al., 2017) and Microsoft Concept Graph (Ji et al., 2019). Notably, we extend the commonsense in two forms: the individual form $\mathcal{C}_1$ and the set form $\mathcal{C}_2$. Both $\mathcal{C}_1$ and $\mathcal{C}_2$ are the sets of triples while each triple in $\mathcal{C}_1$ is constituted of a head entity's concept c_h and a tail entity's concept c_t associated with their instance-level relation r, which can be written as follows:

$$\mathcal{C}_1 = \{(c_h, r, c_t)\} \tag{1}$$

On the contrary, each triple in $\mathcal{C}_2$ consists of a relation r linking the corresponding head concept set C_h and tail concept set C_t, which is shown as:

$$\mathcal{C}_2 = \{(C_h, r, C_t)\} \tag{2}$$

The detailed description of commonsense generation is introduced in section 3.2.

KGE Score Function. We could leverage any KGE model to learn the entity and relation embeddings owing to our scalable framework independent of the KGE model. Thus, we define a uniform symbol $E(h, r, t)$ to represent the score function of any KEG model for evaluating the plausibility of a triple (h, r, t). More specifically, the three most typical score function patterns are given as follows:

(1) The **translation**-based score function, such as TransE (Bordes et al., 2013):

$$E(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \tag{3}$$

where $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ denote the embeddings of head entity h, relation r and tail entity t, respectively.

(2) The **rotation**-based score function, such as RotatE (Sun et al., 2019):

$$E(h, r, t) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| \tag{4}$$

where $\circ$ indicates the hardmard product.

(3) The **tensor decomposition**-based score function, such as DistMult (Yang et al., 2015):

$$E(h, r, t) = \mathbf{h}^\top diag(\mathbf{M}_r)\mathbf{t} \tag{5}$$

where $diag(\mathbf{M}_r)$ represents the diagonal matrix of the relation r.

Link Prediction. Following most of the previous KGC models, we regard link prediction as an entity prediction task. Given a triple query with an entity missing $(h, r, ?)$ or $(?, r, t)$, link prediction takes every entity as a candidate. It calculates the score of each candidate triple by employing the learned KG embeddings and the score function. Then, we rank the candidate entities in light of their scores and output the top n entities as results.

3.2 Automatic Commonsense Generation

In terms of the representation of commonsense defined in section 3.1, our approach could theoretically generate commonsense from any KG automatically as long as there exist some concepts linked to the entities in the KG. Specifically, we develop an entity-to-concept converter to replace the entities in each factual triple with corresponding concepts. Meanwhile, the relations in commonsense entail the instance-level relations in factual KGs. Take an instance in Figure 2, the factual triple $(David, Nationality, U.S.A.)$ can be transformed to a concept-level triple $(Person, Nationality, Country)$. Particularly, the commonsense in the individual form $\mathcal{C}_1$ is achieved by wiping out the reduplicated concept-level triples. Afterwards, we merge the concept-level triples that contain the same relation into a set to construct the commonsense in the set form $\mathcal{C}_2$.
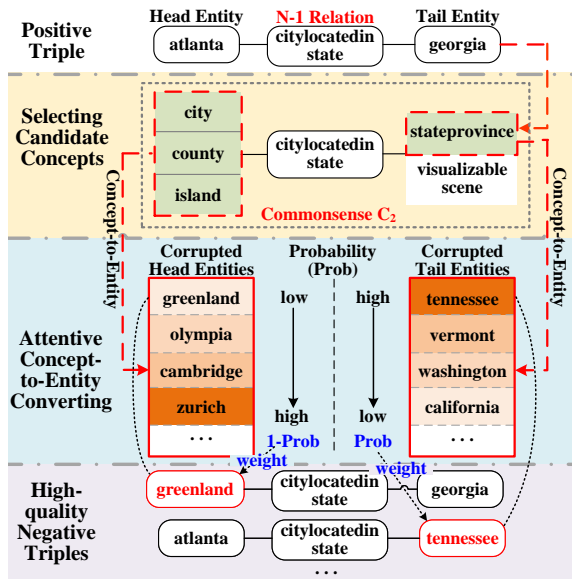
Figure 3: An example of generating the high-quality negative triples containing an N-1 relation by our designed CANS module on NELL-995.

3.3 Commonsense-Aware Negative Sampling

Intuitively, the negative triples satisfying commonsense are more challenging to distinguish from positive triples, contributing to more effective training signals. Therefore, we try to sample the negative triples that conform to the commonsense.

To reduce the false-negative triples, we exploit the characteristics of complex relations, namely 1-1, 1-N, N-1, and N-N defined in TransH (Wang et al., 2014) for negative sampling, where **1** implies that the entity is unique when given the relation and another entity, on the contrary, **N** denotes that there might be multiple entities in this case (non-unique entity). Based on this observation, two specific sampling strategies are proposed: (1) **uniqueness sampling**: in terms of corrupting a unique entity such as the tail entity of the N-1 relation, the corrupted triples except for the original positive one are definitely actual negative triples. Furthermore, the corrupted entities that share at least one concept with the correct entity are regarded as high-quality negative triples, contributing to a more consistent training signal. (2) **None-unique sampling**: for corrupting a non-unique entity such as a head entity linked by the N-1 relation, the entities belonging to the same concept(s) with the correct entity are more likely to be false-negative due to the non-uniqueness of the head entity. Thus, the weights of these negative triples being false-negative should be as low as possible in training. Meanwhile, we

try to sample the triples conforming to the commonsense $\mathcal{C}_2$ for high quality.

For a better understanding, an example of generating high-quality negative triples with an N-1 relation is shown in Figure 3. The whole NS procedure can be divided into two steps. **Step 1**: selecting the candidate concepts with commonsense $\mathcal{C}_2$. The candidate head concepts $city$, $county$ and $island$ are determined according to commonsense $\mathcal{C}_2$ and non-unique sampling. Besides, based on the uniqueness sampling strategy, the candidate tail concept is selected as the same concept $stateprovince$ as that of $Georgia$. **Step 2**: attentive concept-to-entity converting. To reduce false-negative while ensuring the high quality of the negative triples, the corrupted entities belonging to the candidate concepts are sampled from the following distribution:

$$w(h'_j, r, t) = 1 - p((h'_j, r, t)|\{(h_i, r_i, t_i)\})$$
$$= 1 - \frac{\exp \alpha E(h'_j, r, t)}{\sum_i \exp \alpha E(h'_i, r, t)} \quad (6)$$
$$w(h, r, t'_j) = p((h, r, t'_j)|\{(h_i, r_i, t_i)\})$$
$$= \frac{\exp \alpha E(h, r, t'_j)}{\sum_i \exp \alpha E(h, r, t'_i)} \quad (7)$$

where h'_i and t'_i are the corrupted head and tail entities obtained by non-unique sampling and uniqueness sampling. w and p denote the weight and the probability of the negative triple, respectively. α is the temperature of sampling motivated by the self-adversarial sampling (Sun et al., 2019). Remarkably, considering that a triple with a higher probability is more likely to be a positive one, the weight of a negative triple containing the corrupted head entity such is defined as Eq. 6 to prevent the issue of false-negative. Besides, the negative triples containing the corrupted tail entities with higher probability are endowed with higher-quality weight since there is no false-negative issue. Thus, both the corrupted head entity $greenland$ and the corrupted tail entity $tennessee$ with the high weights are selected to generate high-quality negative triples. Similarly, we can generate the high-quality negative triples with an 1-N relation via uniqueness sampling for head entity and none-uniqueness sampling for tail entity. Besides, the negative triples with a 1-1 relation can be obtained by only conducting the uniqueness sampling for both head and tail entities. On the contrary, only none-uniqueness sampling is required for generating negative triples with an N-N relation.

3.4 Traning the KGE Model

Based on the negative triples obtained by CANS, we train the KGE model to learn the entity and relation embeddings for enlarging the gap between the scores of the positive and high-quality negative triples. In this work, we employ the following loss function as our optimization objective:

$$L = -\log \sigma(\gamma - E(h, r, t))$$
$$- \sum_{i=1}^{n} 0.5 \cdot [w(h'_i, r, t) \log \sigma(E(h'_i, r, t) - \gamma) \quad (8)$$
$$+ w(h, r, t'_i) \log \sigma(E(h, r, t'_i) - \gamma)]$$

in which γ is the margin. σ is the sigmoid function.

3.5 Multi-View Link Prediction

Benefiting from the same relations among commonsense and facts, commonsense could directly provide a definite range for link prediction results. Hence we develop a novel multi-view link prediction (MVLK) mechanism in a coarse-to-fine paradigm to facilitate more likely predicted results. Firstly, at the coarse prediction stage, we pick out the candidate entities in the view of commonsense. Specifically, take a query $(h, r, ?)$ for an example, commonsense $\mathcal{C}_1$ is employed for filtering the reasonable concepts of the tail entity. The candidate concept set of tail entity is defined as

$$\mathbf{C}_{1t} = \{c_{ti} | (c_{hi}, r, c_{ti}) \in \mathcal{C}_1\} \quad (9)$$

where c_{hi} is the i-th concept of h, and c_{ti} denotes the tail concept in the commonsense (c_{hi}, r, c_{ti}). Then, the entities belonging to the concept set $\mathbf{C}_{1t}$ can be determined as the candidate entities since they satisfy commonsense and are more likely to be the correct tail entities from the perspective of commonsense compared with other entities.

Then, at the fine prediction stage, we score each candidate entity e_i derived from the coarse prediction stage in the view of fact as following

$$score(e_i) = E(h, r, e_i) \quad (10)$$

where $E(h, r, e_i)$ denotes the score function employed for training the KGE model. Subsequently, the prediction results will rank the scores of candidate entities in ascending order and output the entities with higher ranks.

4 Experiments and Results

In this section, we perform extensive experiments of KGC on four widely-used KG datasets containing concepts. We firstly describe datasets, baseline models, implementation details and evaluation

Dataset	#Rel	#Ent	#Con	#Train	#Valid	#Test
FB15K	1,345	14,951	89	483,142	50,000	59,071
FB15K237	237	14,505	89	272,115	17,535	20,466
NELL-995	200	75,492	270	123,370	15,000	15,838
DBpedia-242	298	99,744	242	592,654	35,851	30,000

Table 1: Statistics of the experimental datasets. #Rel, #Ent, #Con represent the number of relations, entities and concepts of each dataset, respectively.

protocol. Then, the effectiveness of our proposed framework CAKE and each module is demonstrated by compared with several baselines. Furthermore, we conduct extensive experiments, including the ablation study and the case study.

4.1 Experiment Settings

Datasets. Four real-world datasets containing ontological concepts are utilized for experiments, including FB15K (Bordes et al., 2013), FB15K237 (Toutanova and Chen, 2015), NELL-995 (Xiong et al., 2017) and DBpedia-242. Particularly, DBpedia-242 is extracted from DBpedia (Lehmann et al., 2015) which contains totally 242 concepts. The statistics of the datasets are summarized in Table 1. Notably, the entities in FB15K and FB15K237 always belong to more than one concept while each entity in NELL-995 and DBpedia-242 has only one concept.

Baselines. We compare our CAKE model with three state-of-the-art KGE models, including TransE (Bordes et al., 2013), RotatE (Sun et al., 2019) and HAKE (Zhang et al., 2020). Meanwhile, these baselines are also the basic models integrated with our framework. It is unnecessary to use many baselines since the focus of this work is to observe the impact of applying our CAKE framework to original KGE models instead of defeating all the SOTA models. We provide the results of baselines by running their source codes[2] with the suggested parameters. Note that all the existing type-based and ontology-based models are not chosen as baselines since they are specific to a few KGs and cannot work on most of the datasets in our experiment.

Implementation Details. Each complex relation is labelled in the same way as in TransH (Wang et al., 2014). We use Adam optimizer for the training and tune the hyper-parameters of our model by grid search on the validation sets. Specifically, the

---

[2]The codes of TransE and RotatE: https://github.com/DeepGraphLearning/KnowledgeGraphEmbedding. The code of HAKE: https://github.com/MIRALab-USTC/KGE-HAKE.

Models	FB15K					FB15K237				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	35	0.626	0.838	0.723	0.496	195	0.268	0.454	0.298	0.176
TransE+**CANS**	34	0.671	0.864	**0.761**	0.552	**175**	0.298	0.490	0.333	0.203
TransE+**MVLP**	35	0.636	0.839	0.725	0.513	181	0.290	0.476	0.323	0.186
TransE+**CAKE**	**33**	**0.672**	**0.865**	**0.761**	**0.555**	175	**0.301**	**0.493**	**0.335**	**0.206**
RotatE	35	0.657	0.850	0.746	0.537	204	0.269	0.452	0.298	0.179
RotatE+**CANS**	33	0.702	0.877	0.790	0.588	182	0.296	0.486	0.329	0.202
RotatE+**MVLP**	34	0.688	0.860	0.768	0.579	188	0.308	0.493	0.340	0.217
RotatE+**CAKE**	**31**	**0.705**	**0.878**	**0.792**	**0.593**	**181**	**0.318**	**0.511**	**0.354**	**0.223**
HAKE	34	0.690	0.872	0.780	0.574	176	0.306	0.486	0.337	0.216
HAKE+**CANS**	37	0.723	0.882	0.808	0.616	174	0.315	0.501	0.344	0.221
HAKE+**MVLP**	32	0.729	0.890	0.817	0.622	172	0.320	0.508	0.352	0.226
HAKE+**CAKE**	**30**	**0.741**	**0.896**	**0.825**	**0.646**	**170**	**0.321**	**0.515**	**0.355**	**0.227**

Models	DBpedia-242					NELL-995				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	2733	0.242	0.468	0.344	0.100	1081	0.429	0.557	0.477	0.354
TransE+**CANS**	1889	0.287	0.575	0.427	0.103	1022	0.433	0.591	0.495	0.336
TransE+**MVLP**	**881**	0.322	0.585	0.450	0.152	336	0.509	0.617	0.547	0.444
TransE+**CAKE**	**881**	**0.330**	**0.595**	**0.458**	**0.160**	**317**	**0.533**	**0.650**	**0.578**	**0.461**
RotatE	1950	0.374	0.582	0.457	0.249	2077	0.460	0.553	0.493	0.403
RotatE+**CANS**	1063	0.407	0.593	0.476	0.300	1097	0.531	0.644	0.573	0.461
RotatE+**MVLP**	**983**	0.393	0.594	0.474	0.273	356	0.519	0.628	0.564	0.447
RotatE+**CAKE**	1027	**0.423**	**0.603**	**0.486**	**0.320**	**329**	**0.546**	**0.660**	**0.592**	**0.474**
HAKE	1757	0.408	0.579	0.463	0.312	1157	0.502	0.610	0.538	0.437
HAKE+**CANS**	1147	0.427	0.587	0.472	0.341	2011	0.520	0.640	0.556	0.451
HAKE+**MVLP**	1083	0.411	0.580	0.463	0.319	478	0.510	0.614	0.551	0.444
HAKE+**CAKE**	**931**	**0.437**	**0.593**	**0.481**	**0.353**	**433**	**0.543**	**0.655**	**0.583**	**0.477**

Table 2: Link prediction results on four datasets. **Bold** numbers are the best results for each type of model.

embedding size and the batch size are the same as those of each basic model for a fair comparison. The negative sampling size is set as 16 for all the models considering the memory limitation and efficiency. The learning rate is chosen from 0.0001 to 0.01. The margin is tuned in {9, 12, 18, 24, 30}. The sampling temperature is adjusted in {0.5, 1.0}. The entity and relation embeddings are initialized randomly. All the experiments are conducted in Pytorch and on GeForce GTX 2080Ti GPUs.

Evaluation Protocol. Following the procedure of MVLP in Section 3.5, we can obtain the rank of the correct entity for each test example. Then, the performance of link prediction is evaluated by three commonly-used metrics: mean rank (MR), mean reciprocal rank (MRR), and proportion of the correct entities ranked in the top n (Hits@N). All the metrics are in the filtered setting by wiping out the candidate triples already exist in the datasets.

4.2 Experimental Results

Table 2 exhibits the evaluation results of link prediction on the four datasets. We can observe that both CANS and MVLP modules effectively im-

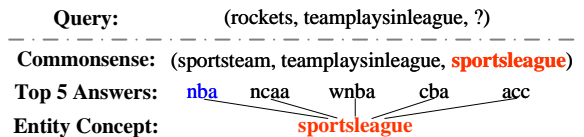| Query: | (rockets, teamplaysinleague, ?) |
Commonsense:	(sportsteam, teamplaysinleague, **sportsleague**)
Top 5 Answers:	nba ncaa wnba cba acc
Entity Concept:	**sportsleague**

Figure 4: A case study of explainable link prediction with commonsense and entity concept on NELL-995.

prove the performance of each basic model on each dataset. Moreover, the entire CAKE framework further facilitates more performance gains than each separate module and outperforms all the baselines consistently and significantly. Compared with the performance average of the three baseline models, our CAKE model improves MRR by **7.2**%, **11.5**%, **16.2**% and **16.7**% on FB15K, FB15K237, DBpedia-242 and NELL-995. These results demonstrate the superiority and effectiveness of integrating commonsense with the original KGE models.

We compare our CANS module with various types of NS techniques, including uniform sampling (Bordes et al., 2013), none sampling (Li et al., 2021), NSCaching (Zhang et al., 2019b), domain-based sampling (Wang et al., 2019b) and

Models	FB15K					FB15K237				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE+Unifo	178	0.301	0.505	0.339	0.201	361	0.171	0.323	0.182	0.097
TransE+NoSamp	144	0.350	0.578	0.415	0.227	343	0.261	0.446	0.297	0.168
TransE+NSCach	209	0.292	0.560	0.375	0.144	556	0.205	0.353	0.226	0.131
TransE+Domain	35	0.619	0.839	0.715	0.489	186	0.283	0.467	0.314	0.190
TransE+SAdv	35	0.626	0.838	0.723	0.496	195	0.268	0.454	0.298	0.176
TransE+CANS (Ours)	**34**	**0.671**	**0.864**	**0.761**	**0.552**	**175**	**0.298**	**0.490**	**0.333**	**0.203**

Models	DBpedia-242					NELL-995				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE+Unifo	5750	0.124	0.262	0.183	0.033	8650	0.167	0.354	0.219	0.068
TransE+None	2292	0.202	0.395	0.247	0.101	9172	0.176	0.297	0.210	0.106
TransE+NSCach	5465	0.156	0.340	0.212	0.050	13967	0.107	0.205	0.122	0.107
TransE+Domain	3415	0.203	0.510	0.346	0.009	1319	0.381	0.549	0.468	0.271
TransE+SAdv	2733	0.242	0.468	0.344	0.100	1081	0.429	0.557	0.477	**0.354**
TransE+CANS (Ours)	**1889**	**0.287**	**0.575**	**0.427**	**0.103**	**1022**	**0.433**	**0.591**	**0.495**	0.336

Table 3: Comparison results of various NS techniques. Unifo, NoSamp, NSCach, Domain and SAdv denote uniform sampling, none sampling, NSCaching, domain-based NS and self-adversarial NS strategies, respectively.

self-adversarial sampling (Sun et al., 2019). The comparison results are obtained by combining these NS techniques[3] with the most classical KGE model TransE(Bordes et al., 2013). From the results shown in Table 3, our CANS module significantly outperforms all the other NS techniques on all the datasets. Specifically, domain-based NS, self-adversarial sampling and our CANS module consistently outperform the others due to the consideration of the quality of negative triples. Furthermore, our CANS module performs better than domain-based NS and self-adversarial sampling since CANS could reduce false-negative. These results illustrate the superior ability of our CANS module to generate more high-quality negative triples for enhancing the performance of any KGE model.

4.3 Ablation Study

We verify the effectiveness of each contribution via integrating the whole framework CAKE and the following ablated models into the basic model HAKE: (1) neglecting the characteristics of complex relations in CANS (-CRNS), (2) removing the commonsense in CANS while retaining the characteristics of complex relations (-CSNS), and (3) omitting the commonsense-view prediction from MVLP (-MVLP). The results in Table 4 demonstrate that our whole model CAKE performs better

Models	FB15K237				
	MR	MRR	Hits@10	Hits@3	Hits@1
CAKE	**170**	**0.321**	**0.515**	**0.355**	**0.226**
-CRNS	186	0.318	0.507	0.352	0.223
-CSNS	182	0.317	0.509	0.351	0.222
-MVLP	174	0.315	0.501	0.344	0.221

Models	NELL-995				
	MR	MRR	Hits@10	Hits@3	Hits@1
CAKE	**433**	**0.543**	**0.655**	**0.583**	**0.477**
-CRNS	650	0.519	0.627	0.564	0.453
-CSNS	447	0.529	0.647	0.567	0.463
-MVLP	2011	0.520	0.640	0.556	0.451

Table 4: Ablation study of integrating each model into the basic model HAKE on FB15K237 and NELL-995.

than all the ablated models on each dataset. It illustrates that introducing commonsense and the characteristics of complex relations both make sense in the NS process for generating more effective negative triples. Besides, MVLP facilitates link prediction performance benefited from determining the reasonable candidate entities by prior commonsense. In general, each contribution plays a pivotal role in our approach.

4.4 Case Study

We provide the case study of explainable link prediction with commonsense as shown in Figure 4. Given a query with the tail entity missing $(rockets, teamplaysinleague, ?)$ on NELL-995, our model could output the answer entities and provide the corresponding entity concepts together with the commonsense specific to the query. We can observe that all the top-

5 entities including the correct entity nba belong to the concept $sportsleague$ which satisfies the commonsense ($rockets, teamplaysinleague, sportsleague$). More interestingly, the commonsense and the entity concepts could explain the rationality of the predicted answer entities to enhance the users' credibility of the answers.

5 Conclusion

In this paper, we propose a novel and scalable commonsense-aware knowledge embedding framework, which could automatically generate commonsense from KGs with entity concepts for the KGC task. We exploit the generated commonsense to produce effective and high-quality negative triples. On the other hand, we design a multi-view link prediction technique in a coarse-to-fine paradigm to filter the candidate entities in the view of commonsense and output the predicted results from the perspective of fact. The experiments on four datasets demonstrate the effectiveness and the scalability of our proposed framework and each module compared with the state-of-the-art baselines. Furthermore, our framework could explain link prediction results and potentially assemble new KGE models to improve their performance.

Acknowledgements

References

Kurt Bollacker, Georg Gottlob, and Sergio Flesca. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Processing of the 27th Annual Conference on Neural Information Processing Systems*, pages 2787–2795.

Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1470–1480.

KrompaßDenis, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *Proceedings of the 14th International Conference on The Semantic Web - ISWC 2015 - Volume 9366*, page 640–655.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 601–610.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2015. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730.

Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1709–1719.

L. Ji, Y. Wang, B. Shi, D. Zhang, Z. Wang, and J. Yan. 2019. Microsoft concept graph: Mining semantic concepts for short text understanding. *Data Intelligence*, 1:262–294.

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 529–539.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, S̈oren Auer, and Christian Bizer. 2015. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Zelong Li, Jianchao Ji, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Chong Chen, and Yongfeng Zhang. 2021. Efficient non-sampling knowledge graph embedding. In *Proceedings of the Web Conference 2021*, page 1727–1736.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253.

Weiyu Liu, Angel Daruna, Zsolt Kira, and Sonia Chernova. 2020. Path ranking with attention to type hierarchies. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 2893–2900.

Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Any-time bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 3137–3143.

T. Mitchell, W. Cohen, E. Hruschka, and et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, page 809–816.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. Drum: End-to-end differentiable rule mining on knowledge graphs. In *Proceedings of the Thirty-third Conference on Neural Information Processing Systems*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3027–3035.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, page 4444–4451.

Yawei Sun, Lingling Zhang, Gong Cheng, and Yuzhong Qu. 2020. SPARQA: skeleton-based semantic parsing for complex questions over knowledge bases. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8952–8959.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of International Conference on Learning Representations*.

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.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, page 2071–2080.

Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019a. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 968–977.

Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference*, pages 878–887.

Yashen Wang, Yifeng Liu, Huanhuan Zhang, and Haiyong Xie. 2019b. Leveraging lexical semantic information for learning concept-based multiple embedding representations for knowledge graph completion. In *APWeb-WAIM*, pages 382–397.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, page 1112–1119.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016a. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 2659–2665.

Ruobing Xie, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2017. Image-embodied knowledge representation learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3140–3146.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016b. Representation learning of knowledge graphs with hierarchical types. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, page 2965–2971.

Wenhan Xiong, Thien Hoang, , and William Yang Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573.

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 *Proceedings of International Conference on Learning Representations*.

Shiquan Yang, Rui Zhang, and Sarah Erfani. 2020. GraphDialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1878–1888.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019a. Quaternion knowledge graph embeddings. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pages 2731–2741.

Yongqi Zhang, Quanming Yao, Yingxia Shao, and Lei Chen. 2019b. Nscaching: Simple and efficient negative sampling for knowledge graph embedding. In *2019 IEEE 35th International Conference on Data Engineering*, pages 614–625.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3065–3072.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4623–4629.