Language Models as Knowledge Embeddings

Xintao Wang[1], **Qianyu He**[1], **Jiaqing Liang**[1*], **Yanghua Xiao**[1,2*]
[1]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[2]Fudan-Aishu Cognitive Intelligence Joint Research Center
{xtwang21, qyhe21}@m.fudan.edu.cn, l.j.q.light@gmail.com, shawyh@fudan.edu.cn

Abstract

Knowledge embeddings (KE) represent a knowledge graph (KG) by embedding entities and relations into continuous vector spaces. Existing methods are mainly structure-based or description-based. Structure-based methods learn representations that preserve the inherent structure of KGs. They cannot well represent abundant long-tail entities in real-world KGs with limited structural information. Description-based methods leverage textual information and language models. Prior approaches in this direction barely outperform structure-based ones, and suffer from problems like expensive negative sampling and restrictive description demand. In this paper, we propose LMKE, which adopts **L**anguage **M**odels to derive **K**nowledge **E**mbeddings, aiming at both enriching representations of long-tail entities and solving problems of prior description-based methods. We formulate description-based KE learning with a contrastive learning framework to improve efficiency in training and evaluation. Experimental results show that LMKE achieves state-of-the-art performance on KE benchmarks of link prediction and triple classification, especially for long-tail entities.

1 Introduction

Knowledge graphs (KGs) are multi-relational graphs composed of entities as nodes and relations as edges, such as WordNet [Miller, 1995]. They have been used to support a wide range of applications, including information retrieval, recommender systems and question answering. For better applications of symbolic knowledge in KGs in recent machine learning models, many efforts have been devoted to embedding KGs into low-dimension vector spaces, which are referred to as knowledge embeddings (KEs). The principal applications of KEs are link prediction and triple classification in KGs, while there is also an increasing trend to use KEs in natural language processing (NLP) tasks like text generation.

There are mainly two kinds of existing KE methods, namely traditional structure-based methods and emerging description-based methods. Structure-based methods, such as TransE [Bordes *et al.*, 2013] and RotatE [Sun *et al.*, 2019], are trained to preserve the inherent structure of KGs. These methods cannot well represent long-tail entities, as they depend solely on the structure of KGs and thus favor entities rich in structure information (i.e., linked with plentiful entities). However, real-world KGs are widely observed to have a right-skewed degree distribution, i.e., degrees of entities approximately follow the power-law distribution that forms a long tail, as is illustrated in Fig.1. These KGs are populated with massive unpopular entities of low degrees. For example, on WN18RR, 14.1% entities are with degree 1, and 60.7% entities have no more than 3 neighbors. Their embeddings consequently suffer from limited structural connectivity. This problem is justified by the declined performance of structure-based methods on long-tail entities shown in Fig.1, which suggests that their embeddings are still unsatisfactory.

Description-based KE methods represent entities in KGs by encoding their descriptions with language models, such as DKRL [Xie *et al.*, 2016] and KEPLER [Wang *et al.*, 2019b]. Textual descriptions provide rich information for numerous semantic-related tasks, which brings an opportunity to learn informative representations for long-tail entities. In an extreme case, an emerging entity that is novel to an existing KG (in other words, having zero degrees in the KG) can still be well represented with its textual information. This feature is referred to as a capacity in the “inductive (zero-shot) setting” by prior approaches. Besides, lots of missing knowledge in KGs could be covered by rich textual information contained in entity descriptions or learned by pretrained language mod-

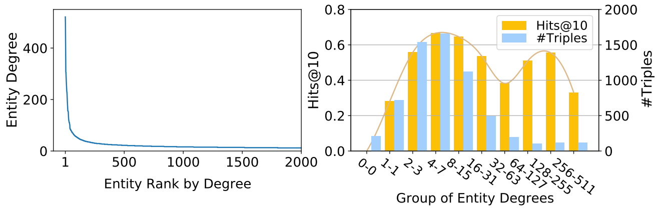

Figure 1: Distribution of entity degrees and performance of RotatE (a typical structure-based method) on WN18RR. Entities are grouped by the logarithm of degrees (right). For each group, we report the number of relevant triples and their average performance on link prediction. Please refer to Sec 4.3 for more details.

*Corresponding Authors

els (PLMs), given that current text corpora outstrip KGs in scale and contain much more information. However, existing description-based methods barely outperform structure-based methods and suffer from the following problems:

1. *Expensive negative sampling*. While negative sampling is vital to KE learning, existing description-based methods are only allowed to have limited negative samples because of the encoding expense of language models.
2. *Restrictive description demand*. Existing methods generally require descriptions of all entities in the KG, and discard entities with no or short descriptions. Although delicate benchmarks can satisfy this demand, real-world KGs can hardly contain descriptions for all entities.

In this paper, we propose LMKE, which adopts **L**anguage **M**odels to derive **K**nowledge **E**mbeddings, aiming at both enhancing representations of long-tail entities and solving the above issues in prior description-based KEs. LMKE leverages the inductive capacity of description-based methods to enrich the representations of long-tail entities. In LMKE, entities and relations are regarded as special tokens whose output embeddings are contextualized in and learned from related entities, relations, and their textual information, so LMKE is also suitable for entities with no description. A contrastive learning framework is further proposed where entity embeddings within a mini-batch serve as negative samples for each other to avoid the additional cost of encoding negative samples. In summary, our contributions are listed as follows:[1]

1. We identify the problem of structure-based KEs in representing long-tail entities, and generalize the inductive capacity of description-based KEs to solve this problem. To the best of our knowledge, we are the first to propose leveraging textual information and language models to enrich representations of long-tail entities.
2. We propose a novel method LMKE that tackles two deficiencies of existing description-based methods, namely expensive negative sampling and restrictive description demand. We are also the first to formulate description-based KE learning as a contrastive learning problem.
3. We conduct extensive experiments on widely-used KE benchmarks, and show that LMKE achieves the state-of-the-art performance on both link prediction and triple classification, significantly surpassing existing structure-based and description-based methods, especially for long-tail entities.

2 Related Work

Knowledge embeddings represent entities and relations of KGs in low-dimension vector spaces. KGs are composed of triples, where a triple (h, r, t) means there is a relation $r \in \mathcal{R}$ between the head entity $h \in \mathcal{E}$ and the tail entity $t \in \mathcal{E}$. $\mathcal{E}$ and $\mathcal{R}$ denote the entity set and the relation set respectively.

Structure-Based Knowledge Embeddings. Existing KEs are mainly structure-based, deriving the embeddings by solely structure information of KGs. These methods are further distinguished into translation-based models and semantic matching models in terms of their scoring functions [Wang *et al.*, 2017]. Translation-based models adopt distance-based scoring functions, which measure the plausibility of a triple (h, r, t) by the distance between entity embeddings $\mathbf{h}$ and $\mathbf{t}$ after a translation specific to the relation. The most representative model is TransE. It embeds entities and relations as $\mathbf{h}$, $\mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ in a shared vector space with a dimension d. Its loss function is defined as $||\mathbf{h} + \mathbf{r} - \mathbf{t}||$ so as to make $\mathbf{h}$ close to $\mathbf{t}$ after a translation $\mathbf{r}$. TransH [Wang *et al.*, 2014] proposes to project entity embeddings $\mathbf{h}$ and $\mathbf{t}$ to a relation-specific hyperplane, and TransR [Lin *et al.*, 2015] further proposes projections into relation-specific spaces. RotatE defines a relation as a rotation from entity h to entity t in a complex vector space, so their embeddings $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t} \in \mathbb{C}^d$ are expected to satisfy $\mathbf{h} \odot \mathbf{r} \approx \mathbf{t}$, where $\odot$ stands for element-wise product. Semantic matching models adopt similarity-based scoring functions, which measure the plausibility of a triple (h, r, t) by matching latent semantics of h, r, t. RESCAL [Nickel *et al.*, 2011] represents the relation r as a matrix $\mathbf{M}_r$ and use a bilinear function $\mathbf{h}^\top \mathbf{M}_r \mathbf{t}$ to score (h, r, t). DistMult [Yang *et al.*, 2014] makes $\mathbf{M}_r$ diagonal for simplicity and efficiency. CoKE [Wang *et al.*, 2019a] employs Transformers [Vaswani *et al.*, 2017] to derive contextualized embeddings, where triples or relation paths serve as the input token sequences.

Description-based Knowledge Embeddings. Recently, description-based KE methods are gaining increasing attention for the importance of textual information and development in NLP. DKRL [Xie *et al.*, 2016] first introduces entities' descriptions and encodes them via a convolutional neural network for description-based embeddings. KEPLER [Wang *et al.*, 2019b] uses PLMs as the encoder to derive description-based embeddings and is trained with the objectives of both KE and PLM. Pretrain-KGE [Zhang *et al.*, 2020b] proposes a general description-based KE framework, which initializes another learnable KE with description-based embeddings and discards PLMs for efficiency after fine-tuning PLMs. KG-BERT [Yao *et al.*, 2019] concatenates descriptions of h, r, t as an input sequence to PLMs, and scores the triple by the sequence embedding. StAR [Wang *et al.*, 2020] thus partitions a triple into two asymmetric parts between which it performs semantic matching.

3 Methods

In this section, we introduce LMKE and its variants. We first provide background on language models (Sec 3.1). Afterward, we elaborate on how LMKE adopts language models to derive knowledge embeddings (Sec 3.2), and its contrastive variants for both zero-cost negative sampling in training and efficient link prediction in evaluation (Sec 3.3).

3.1 Language Models

Pretrained language models have been increasingly prevalent in NLP. They have been pretrained on large-scale corpora to store vast amounts of general knowledge. For example, BERT [Devlin *et al.*, 2018] is a Transformer encoder pretrained to predict randomly masked tokens. Afterward, PLMs

[1]Our codes are available at https://github.com/Neph0s/LMKE

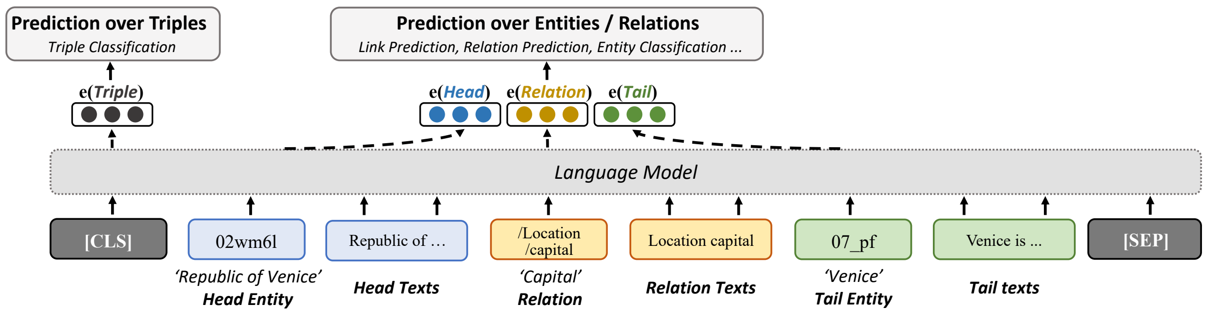

Figure 2: The architecture of LMKE.

can be readily used to achieve excellent performance in various downstream tasks with fine-tuning.

To better understand such excellence, knowledge probing is proposed [Petroni *et al.*, 2019], which questions PLMs with masked cloze sentences. Researches in this direction have demonstrated that PLMs contain abundant factual knowledge and have the potential to be used as knowledge bases. Interestingly, PLMs are also shown to be capable of learning relations satisfying one-hop rules like equivalence, symmetry, inversion, and implication [Kassner *et al.*, 2020], which is also the desiderata of knowledge embeddings.

3.2 LMKE

LMKE adopts language models as knowledge embeddings, i.e., deriving embeddings of entities and relations that can support prediction of the plausibility of given triples.

LMKE learns embeddings of entities and relations in the same space as word tokens. Given a specific triple $u = (h, r, t)$, LMKE leverages descriptions s_h, s_r, s_t of h, r, t as additional input, and outputs its probability $p(u)$. LMKE makes every entity and relation a token and gets their embeddings via pretrained language models. The triple is transformed into a sequence of tokens of h, r, t and the words in their descriptions, which serves as the input to PLMs. The description of an entity (or relation) e is a sequence of tokens $s_e = (x_1, \dots, x_{n_e})$ where n_e denotes the length. The input sequence is thus $s_u = (h, s_h, r, s_r, t, s_t) = (h, x_1^h, \dots, x_{n_h}^h, r, x_1^r, \dots, x_{n_r}^r, t, x_1^t, \dots, x_{n_t}^t)$. Then, two special tokens [CLS] and [SEP] are inserted in the front and back of s_u. Feeding forward s_u into the PLM, LMKE generates the encoded embeddings $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ of the three tokens h, r, t. In this way, we embed entities, relations, and words of their descriptions in a shared vector space. The embedding of one entity (or relation) is contextualized in and learned from not only its own textual information but also the other two components of the triple and their textual information. Therefore, long-tail entities can be well represented with their descriptions, and entities without descriptions can also learn representations from related entities' descriptions. The output embedding of [CLS] aggregates information of the whole sequence, so we regard it as embedding $\mathbf{u}$ for the entire triple. To score plausibility $p(u)$ of the triple, LMKE inputs $\mathbf{u}$ into a linear layer like KG-BERT:

$$p(u) = \sigma(\mathbf{w}\mathbf{u} + b) \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ and b are learnable weight and bias. We adopt binary cross-entropy as the loss function for training. Besides, LMKE can also be used for prediction over entities or relations with $\mathbf{h}, \mathbf{r}, \mathbf{t}$. The architecture is shown in Fig. 2.

The principal applications of KEs are predicting missing links and classifying possible triples, which are formulated as two benchmark tasks for KE evaluation, namely ***link prediction*** [Bordes *et al.*, 2013] and ***triple classification*** [Socher *et al.*, 2013]. Triple classification is a binary classification task that judges whether a triple u is true or not, which can be directly performed with $p(u)$. Link prediction is to predict the missing entity in a corrupted triple $(?, r, t)$ or $(h, r, ?)$ where $?$ denotes an entity removed for prediction. In this task, models need to corrupt a triple by replacing its head or tail entity with every entity in the KG, score the replaced triples, and rank the entities in descending order of the scores. However, it is hardly affordable for LMKE to score every replaced triple as an integral with PLMs, concerning the time complexity shown in Table 1. Even for a middle-sized dataset FB15k-237, there would be 254 million triples to be encoded.

Method	Training	Link Prediction
KG-BERT	$O(N_{\text{train}} N_{\text{neg}})$	$O(\lvert\mathcal{E}\rvert N_{\text{eval}})$
StAR	$O(N_{\text{train}} N_{\text{neg}})$	$O(\lvert\mathcal{E}\rvert\lvert\mathcal{R}\rvert)$
LMKE	$O(N_{\text{train}} N_{\text{neg}})$	$O(\lvert\mathcal{E}\rvert N_{\text{eval}})$
C-LMKE	$O(N_{\text{train}})$	$O(N_{\text{eval}} + \lvert\mathcal{E}\rvert)$

Table 1: The time complexity of training and link prediction evaluation. $|\mathcal{E}|$ or $|\mathcal{R}|$ denotes the number of entities or relations in the KG. N_{train} or N_{eval} is the number of triples in the training or evaluation split. N_{neg} is the negative sampling size. C-LMKE denotes contrastive LMKE, whose complexity is lower than prior approaches.

3.3 Contrastive KE Learning

Towards efficient link prediction with language models, one solution is to encode the triples partially. Masked entity model (MEM-KGC) [Choi *et al.*, 2021] replaces the removed entity and its description with a mask q, and predicts the missing entity by feeding its output embedding $\mathbf{q}$ into a linear layer. It can be viewed as a masked variant of LMKE that trades off the exploitation of textual information for time complexity. Since only one masked incomplete triple is encoded, the complexity is downgraded. Nevertheless, it drops textual information of the target entities to be predicted, thus hurting the utility of textual information.

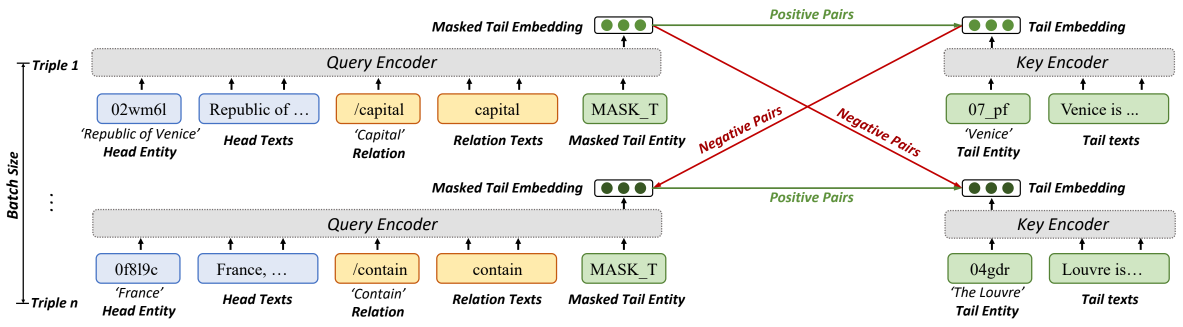

Figure 3: The architecture of contrastive LMKE. The queries and keys are encoded separately before contrastive matching within the batch.

We propose a contrastive learning framework to better exploit description-based KEs for link prediction, where the *given entity-relation pair* and the *target entity* serve as the query q and the key k to be matched in contrastive learning.

From this point of view, the output embedding $\mathbf{q}$ of the masked entity in MEM-KGC is the encoded query, and the i-th row in the weights $\mathbf{W}_e$ of the linear layer serves as the key that corresponds to the i-th entity for each $1 \leq i \leq |\mathcal{E}|$. Therefore, feeding $\mathbf{q}$ into the linear layer can be viewed as matching the query q with the keys. The difference is that the keys are directly learned representations instead of encodings of textual information like the queries.

Contrastive LMKE (C-LMKE) is an LMKE variant under this framework that replaces the learned entity representations (rows of $\mathbf{W}_e$) with encodings of target entities' descriptions, as is shown in Fig. 3. It features *the contrastive matching within the mini-batch*, which allows efficient link prediction without loss of information exploitation while avoiding the additional cost of encoding negative samples. The candidate keys of a query are defined as keys of the queries in the batch. C-LMKE's time complexity is analyzed in Table 1.

This practice solves the problem of expensive negative sampling and allows description-based KEs to learn from much more negative samples. While negative samples are vital to KE learning, most existing description-based methods are allowed to have only a few for each positive sample (usually ranging from 1 to 5) because of the cost of language models. C-LMKE currently binds the negative sampling size to the batch size, and with our contrastive framework, existing approaches in contrastive learning like memory bank can be further introduced to achieve more improvement.

We match an encoded query $\mathbf{q}$ with an encoded key $\mathbf{k}$ by a two-layer MLP (multi-layer perceptron), instead of cosine similarity that is commonly adopted in contrastive learning, because there may exist multiple keys matching q. If both $\mathbf{k}_1$ and $\mathbf{k}_2$ match $\mathbf{q}$ and we maximize similarity between $(\mathbf{q}, \mathbf{k}_1)$ and $(\mathbf{q}, \mathbf{k}_2)$, $(\mathbf{k}_1, \mathbf{k}_2)$ will also be enforced to be similar, which is not desirable. Thus, the probability that q matches k is:

$$p(q, k) = \sigma(\mathrm{MLP}[\mathbf{q}; \mathbf{k}; \mathbf{q} - \mathbf{k}; \mathbf{q} \odot \mathbf{k}; \mathbf{d}]) \tag{2}$$

where $\mathbf{d} = [\log(d_q + 1); \log(d_k + 1)]$ is the logarithm of entity degrees. d_q and d_k are the degrees of the given entity in q and the target entity k counted on the training set. If the training set and the test set follow the same distribution, entities of higher degrees will also be more likely to appear in test triples, so degrees are important structural information to be concerned. While structure-based KEs learn degree information as aggregation into clusters [Pei *et al.*, 2019] and MEM-KGC learns it as the imbalance of entity labels, our matching function cannot capture this information, so we explicitly include it as additional features for prediction.

Since there are usually multiple correct entities for a corrupted triple regarding relations that are not one-to-one, we adopt binary cross-entropy to judge whether each entity is a correct key (multi-label classification), instead of multi-class cross-entropy to find the most likely entity. Given that most of the keys are negative, we average losses over positive ones and negative ones respectively, and sum them up. The loss of matching a query q with the keys $\mathcal{K}$ is thus formulated as $L(q, \mathcal{K}) = -\sum_{k^+ \in \mathcal{K}^+} w_{q,k^+} \log(p(q, k^+)) - \sum_{k^- \in \mathcal{K}^-} w_{q,k^-} \log(1 - p(q, k^-))$, where $\mathcal{K}^+, \mathcal{K}^-$ denotes the positive and negative keys respectively and $\mathcal{K} = \mathcal{K}^+ \cup \mathcal{K}^-$. We adopt self-adversarial negative sampling [Sun *et al.*, 2019] for efficient KE learning, which calculates the weights as $w_{q,k^+} = \frac{1}{|\mathcal{K}^+|}$ and $w_{q,k^-} = \frac{\exp(p(q,k^-))}{\sum_{k \in \mathcal{K}^-} \exp(p(q,k))}$. In this case, loss of false-positive samples is amplified and loss of true-negative samples is diminished.

4 Experiments and Analyses

In this section, we evaluate the effectiveness of our methods.

4.1 Experiment Setup

Datasets. We experiment on four popular benchmark datasets: FB13 [Socher *et al.*, 2013], FB15k-237 [Toutanova, 2015], UMLS [Dettmers *et al.*, 2018] and WN18RR [Dettmers *et al.*, 2018], whose statistics are shown in Table 3. FB13 and FB15k-237 are derived from Freebase. WN18RR is derived from WordNet. UMLS is a medical ontology describing relations between medical concepts. FB15k-237 and WN18RR are used for link prediction, where abundant inverse relations are removed in case they can serve as shortcuts. For triple classification, FB13 and UMLS are used. Only the test split of FB13 contains negative samples. In other situations, negative samples are created by randomly replacing head or tail entities, where the ground truth (training triples for the training split and all triples for the test split) would be avoided. Following KG-BERT, the entity descriptions we use are synsets definitions for WN18RR,

Dataset	FB15k-237					WN18RR				
Metric	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
Structure-based Knowledge Embeddings										
TransE [Bordes *et al.*, 2013]♣	323	0.279	0.198	0.376	0.441	2300	0.243	0.043	0.441	0.532
DistMult [Yang *et al.*, 2014]♠	254	0.241	0.155	0.263	0.419	5110	0.430	0.390	0.440	0.490
ComplEx [Trouillon *et al.*, 2016]♠	339	0.247	0.158	0.275	0.428	5261	0.440	0.410	0.460	0.510
RotatE [Sun *et al.*, 2019]♠	177	0.338	0.241	0.375	0.533	3340	0.476	0.428	0.492	0.571
TuckER [Balažević *et al.*, 2019]	-	0.358	0.266	0.394	0.544	-	0.470	0.443	0.482	0.526
HAKE [Zhang *et al.*, 2020a]	-	0.346	0.250	0.381	0.542	-	0.497	0.452	0.516	0.582
CoKE [Wang *et al.*, 2019a]	-	0.364	0.272	0.400	0.549	-	0.484	0.450	0.496	0.553
Description-based Knowledge Embeddings										
Pretrain-KGE$_{\text{TransE}}$ [Zhang *et al.*, 2020b]	162	0.332	-	-	0.529	1747	0.235	-	-	0.557
KG-BERT [Yao *et al.*, 2019]♣	153	-	-	-	0.420	97	0.216	0.041	0.302	0.524
StAR$_{\text{BERT-base}}$ [Wang *et al.*, 2020]	136	0.263	0.171	0.287	0.452	99	0.364	0.222	0.436	0.647
MEM-KGC$_{\text{BERT-base}}$(w/o EP)	-	0.339	0.249	0.372	0.522	-	0.533	0.473	0.570	0.636
MEM-KGC$_{\text{BERT-base}}$(w/ EP)	-	0.346	0.253	0.381	0.531	-	0.557	0.475	0.604	0.704
C-LMKE$_{\text{BERT-tiny}}$	**132**	**0.406**	0.319	**0.445**	**0.571**	148	0.545	0.467	0.587	0.692
C-LMKE$_{\text{BERT-base}}$	183	0.404	**0.324**	0.439	0.556	**72**	**0.598**	**0.480**	**0.675**	**0.806**

Table 2: Results of link prediction on FB15k-237 and WN18RR. ♠ denotes results from [Sun *et al.*, 2019]. ♣ denotes results from [Wang *et al.*, 2020]. We implement StAR on FB15k-237 with BERT-base as the base model. Other results are taken from their original papers. EP denotes the entity prediction task of MEM-KGC. C-LMKE denotes contrastive LMKE.

and descriptions from Wikipedia for FB13, from [Xie *et al.*, 2016] for FB15k-237, and from [Yao *et al.*, 2019] for UMLS. The relation descriptions are their names for all datasets.

Dataset	#Entity	#Relation	#Train	#Dev	#Test	Avg DL
FB13	75,043	13	316,232	5,908	23,733	110.7
FB15k-237	14,541	237	272,115	17,535	20,466	141.7
UMLS	135	46	5,216	652	661	161.7
WN18RR	40,943	11	86,835	3,034	3,134	14.4

Table 3: Statistics of the datasets. Avg DL means the average length (number of words) of descriptions.

Baselines. We compare our methods with structure-based and description-based methods. The structure-based methods include TransE, TransH, TransR, DistMult, ComplEx, RotatE. The description-based methods include Pretrain-KGE, KG-BERT, exBERT, and StAR. For a fair comparison, we reimplement StAR on FB15k-237 with BERT-base.

Metrics. For triple classification, we report accuracy. For link prediction, we report Mean Rank (MR), Mean Reciprocal Rank (MRR), and Hits@$\{1, 3, 10\}$ in the "filtered" setting. Metrics for link prediction are based on the rank of the correct entity in a list of all entities ranked by their plausibility. The "filtered" setting is a common practice that removes other correct entities (which also constitute triples existing in the KG) from the list. Hits@K measures the proportion of correct entities ranked in the top K. The results are averaged over test triples and over predicting missing head and tail entities. Generally, a good model is expected to achieve higher MRR, Hits@N, and lower MR.

Settings. We evaluate LMKE on triple classification and C-LMKE on link prediction with BERT-base [Devlin *et al.*, 2018] and BERT-tiny [Turc *et al.*, 2019] as the language model. Larger models are not considered, in which case we have to use a tiny batch size. We search these hyperparameters with BERT-tiny: learning rate of PLMs among $\{10^{-4}, 5\times10^{-5}, 10^{-5}\}$, learning rate of other components among $\{10^{-3}, 5\times10^{-4}, 10^{-4}, 10^{-5}\}$, batch size among $\{12, 16, 32, 64\}$ based on best Hits@10 on the dev set. With BERT-base, we set the batch size as 16 for triple classification and 12 for link prediction, which considers both results of BERT-tiny and limited memory. Our models are fine-tuned with Adam as the optimizer. For triple classification, we sample 1 negative triple for each positive triple. For link prediction, we encode given entity-relation pair queries and target entity keys with two language models. They are equally initialized and share the same embeddings of words, entities, and relations.

4.2 General Performance

We compare our methods with prior approaches on both link prediction and triple classification. Experimental results in Table 2 and Table 4 show that our methods achieve the state-of-the-art performance on both tasks.

Results of link prediction on FB15k-237 and WN18RR are demonstrated in Table 2, which show that our method significantly outperforms existing approaches. C-LMKE with BERT-base achieves surpassing performance on MR, MRR and Hits@$\{1, 3, 10\}$ on both datasets. The improvement is more significant on WN18RR, where Hits@10 of our method reaches to 0.806 while the best result of previous methods is 0.704 from MEM-KGC. However, its entity prediction task is compatible with our work, without which its Hits@10 declines to 0.636. Our method is also the first description-based one to outperform state-of-the-art structure-based methods on FB15k-237. Even with BERT-tiny, our method achieves better or comparable performance compared with prior approaches built with larger models.

The results suggest that description-based methods largely outperform structure-based ones on WN18RR, but barely surpass them on FB15k-237. We analyze the difference between these datasets to explain this phenomenon. FB15k-237 differs from WN18RR mainly in two aspects: sparsity and description. According to statistics shown in Table 3, the average degree on FB15k-237 and WN18RR is 37.4 and 4.2 respectively. The former is about 8.9 times the latter, which indicates that entities in FB15k-237 generally have access to rich structural information while entities in WN18RR are more likely to be long-tail. Also, textual information better covers structural information on WN18RR compared with FB15k-237. Entities on WN18RR are words, and descriptions are exactly their definitions from which structural relations are derived, so descriptions can ideally support the inference of structural relations. However, entities on FB15k-237 are real-world entities whose collected descriptions only partially support the inference. For example, the fact (*Albert Einstein, isA, peace activist*) is not covered by collected descriptions of these entities. As a result, the overreliance on textual information may hurt performance, and description-based methods did not achieve surpassing performance. This also explains why replacing BERT-base in C-LMKE with BERT-tiny does not decrease the performance.

Dataset	FB13	UMLS
TransE [Bordes *et al.*, 2013]	81.5	78.1
TransH [Wang *et al.*, 2014]	83.3	79.2
TransR [Lin *et al.*, 2015]	82.5	81.9
DistMult [Yang *et al.*, 2014]	86.2	86.8
KG-BERT [Yao *et al.*, 2019]	90.4	89.7
exBERT [Yaser Jaradeh *et al.*, 2021]	-	90.3
LMKE$_{\text{BERT-base}}$	**91.7**	**92.4**

Table 4: Accuracy of triple classification on FB13 and UMLS. Results of existing baselines on FB13 and UMLS are taken from [Yao *et al.*, 2019] and [Yaser Jaradeh *et al.*, 2021] respectively.

Results of triple classification on FB13 and UMLS in Table 4 show that LMKE also outperforms existing methods on this task. Comparison between results of LMKE and KG-BERT demonstrates the effectiveness of learning embeddings of entity and relation tokens in the same space as word tokens.

4.3 Performance Grouped by Entity Degrees

To show the effectiveness of our methods on long-tail entities, we group entities by the logarithm of degrees, collect relevant triples for each group, and study the average link prediction performance of different methods on different groups. A triple (h, r, t) is relevant to group i if h or t is in group i. The grouping rule is the same as in Fig.1. The results on FB15k-237 in Fig.4 show that description-based methods significantly outperform structure-based ones on long-tail entities in group 0, 1 and 2 (whose degrees are lower than 4), and our C-LMKE significantly surpasses other description-based ones. Comparison between results of C-LMKE with or without degree information indicates that introducing degree information generally improves its performance on entities

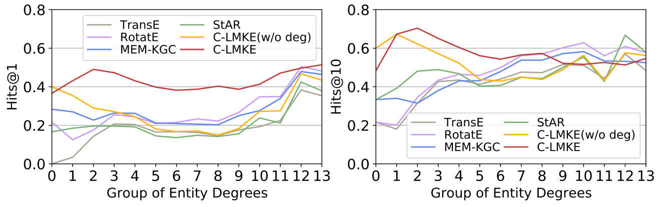

Figure 4: Average Hits@1 and Hits@10 performance grouped by the logarithm of entity degrees on FB15k-237.

that are not long-tail. On popular entities, however, structure-based methods generally perform better. Although StAR is also description-based, it achieves the best Hits@10 on group 12 and 13, because it is trained with an additional objective that follows structure-based methods.

4.4 Importance of Negative Sampling Size

We study the performance of C-LMKE with different negative sampling sizes N_{neg} to justify its importance. We set the batch size as 32 and limit N_{neg} for each triple by only using several encoded target entities of other triples in the batch as negative keys. We report Hits@10 of C-LMKE with BERT-tiny on FB15k-237 for 40 epochs. The results shown in Fig. 5 indicate that larger N_{neg} continuously brings better performance until convergence. Increasing N_{neg} greatly accelerates training, and improves the eventual performance when N_{neg} is under 8. However, prior description-based methods generally set N_{neg} as only 1 or 5, which limits their performance.

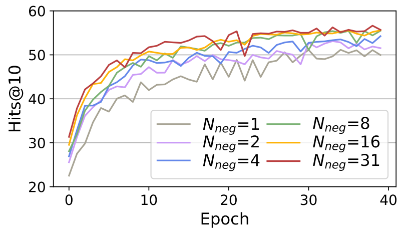

Figure 5: Hits@10 of C-LMKE with BERT-tiny on FB15k-237 with different negative sampling size.

5 Conclusion

In this paper, we propose LMKE, an effective and efficient method to adopt language models as knowledge embeddings. Motivated by the inability of structure-based KEs to well represent long-tail entities, LMKE leverages textual descriptions and learns embeddings of entities and relations in the same space as word tokens. It solves the restrictive demand of prior description-based approaches. A contrastive learning framework is further proposed that allows zero-cost negative sampling and significantly downgrades time complexity in both training and evaluation. Results of extensive experiments show that our methods achieve state-of-the-art performance on various benchmarks, especially for long-tail entities.

In the future, we plan to explore the effectiveness of more advanced contrastive learning approaches in description-based KEs. We are also interested in language models' capacity of modeling composition patterns in KGs.

Acknowledgements

This work was supported by National Key Research and Development Project (No.2020AAA0109302), Shanghai Science and Technology Innovation Action Plan (No.19511120400), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103), China Postdoctoral Science Foundation (No. 2020M681173 and No. 2021T140124), and National Natural Science Foundation of China (Grant No. 62102095).

References

[Balažević *et al.*, 2019] Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*, 2019.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proc. of NeurIPS*, 2013.

[Choi *et al.*, 2021] Bonggeun Choi, Daesik Jang, and Youngjoong Ko. Mem-kgc: Masked entity model for knowledge graph completion with pre-trained language model. *IEEE Access*, 2021.

[Dettmers *et al.*, 2018] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proc. of AAAI*, 2018.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Kassner *et al.*, 2020] Nora Kassner, Benno Krojer, and Hinrich Schütze. Are pretrained language models symbolic reasoners over knowledge? In *Proc. of CoNLL*, 2020.

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

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.

[Nickel *et al.*, 2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.

[Pei *et al.*, 2019] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *Proc. of WWW*, 2019.

[Petroni *et al.*, 2019] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[Socher *et al.*, 2013] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, 2013.

[Sun *et al.*, 2019] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[Toutanova, 2015] Kristina Toutanova. Observed versus latent features for knowledge base and text inference. *ACL-IJCNLP 2015*, page 57, 2015.

[Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proc. of ICML*, 2016.

[Turc *et al.*, 2019] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

[Wang *et al.*, 2014] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proc. of AAAI*, 2014.

[Wang *et al.*, 2017] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2017.

[Wang *et al.*, 2019a] Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. Coke: Contextualized knowledge graph embedding. *arXiv preprint arXiv:1911.02168*, 2019.

[Wang *et al.*, 2019b] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *arXiv preprint arXiv:1911.06136*, 2019.

[Wang *et al.*, 2020] Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, and Yi Chang. Structure-augmented text representation learning for efficient knowledge graph completion. *arXiv preprint arXiv:2004.14781*, 2020.

[Xie *et al.*, 2016] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proc. of AAAI*, 2016.

[Yang *et al.*, 2014] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[Yao *et al.*, 2019] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[Yaser Jaradeh *et al.*, 2021] Mohamad Yaser Jaradeh, Kuldeep Singh, Markus Stocker, and Sören Auer. Triple classification for scholarly knowledge graph completion. *arXiv preprint arXiv:2111.11845*, 2021.

[Zhang *et al.*, 2020a] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proc. of AAAI*, 2020.

[Zhang *et al.*, 2020b] Zhiyuan Zhang, Xiaoqian Liu, Yi Zhang, Qi Su, Xu Sun, and Bin He. Pretrain-kge: Learning knowledge representation from pretrained language models. In *Proc. of EMNLP*, 2020.