Is Knowledge Embedding Fully Exploited in Language Understanding? An Empirical Study

Anonymous ACL submission

Abstract

The recent development of knowledge embedding (KE) enables machines to represent knowledge graphs (KGs) with low-dimensional embeddings, which facilitates utilizing KGs for various downstream natural language understanding (NLU) tasks. However, less work has been done on systematically evaluating the impact of KE on NLU. In this work, we conduct a comprehensive analysis of utilizing KE on four downstream knowledge-driven NLU tasks using two representative knowledge-guided frameworks, including knowledge augmentation and knowledge attention. From the experimental results, we find that: (1) KE models that have better performance on knowledge graph completion do not necessarily help knowledge-driven NLU tasks better in the knowledge-guided frameworks; (2) KE could effectively benefit NLU tasks from two aspects including entity similarity and entity relation information; (3) KE could further benefit pre-trained language models which have already learned rich knowledge from pre-training. We hope the results could help and guide future studies to utilize KE in NLU tasks. Our source code will be released to support further exploration.

1 Introduction

Knowledge graphs (KGs) organize entity knowledge and concept knowledge into structured relational data, potentially providing rich information for a variety of NLP tasks, such as information retrieval (Hu et al., 2009), information extraction (Hoffmann et al., 2011), and question answering (Bordes et al., 2014a,c). Both the research community and the industry have built various large-scale KGs[1] and intend to exploit the rich information in KGs to help natural language understanding.

KG is a typical kind of non-Euclidean data, which is difficult for deep learning models to use directly (Bronstein et al., 2017), while deep learning has become the standard technique of NLP. Knowledge embedding (KE) represents entities and relations in KGs as low-dimensional semantic embeddings in a Euclidean space, which clears the way for injecting KGs into deep learning models. Recently, many efforts have been devoted to KE (Minervini et al., 2017; Guo et al., 2018; Padia et al., 2019) and KE has shown its strong ability to represent knowledge. Hence, it is feasible to integrate KE in downstream NLP tasks.

Although some recent work has explored utilizing KE for NLP, these studies usually only focus on a single task with a single KE (Weston et al., 2013; Bordes et al., 2014a; Zhang et al., 2016; Xin et al., 2018). Less work has been done to systematically evaluate the impact of KE on NLP. To advance the utilization of KE, we need to understand how and to what extent KE contributes to downstream NLP tasks.

In this paper, we focus on the impact of KE on language understanding. First, we summarize two mainstream knowledge-guided frameworks based on existing work: knowledge augmentation and knowledge attention. Then, we perform a comprehensive analysis of utilizing KE on four knowledge-driven NLU tasks. Specifically, we evaluate these frameworks on the following two types of tasks: (1) **Entity-oriented tasks**: relation extraction and entity typing; (2) **General NLU tasks**: information retrieval and fact verification. Besides, to investigate KE's effect with various text encoders, we implement three representative text encoders: CNN (Kim, 2014), LSTM (Hochreiter and Schmidhuber, 1997), and BERT (Devlin et al., 2019). We have the following observations:

(1) In most cases, KE models can improve the performance of the models that only use texts. However, a KE model with better performance on KGC does not necessarily better help NLU tasks.

(2) For what information of KE could help lan-

[1] E.g., YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), and Wikidata (Vrandečić and Krötzsch, 2014).

guage understanding, our experiments show that the models using entity embeddings as external knowledge could effectively capture entity similarity and entity relation information.

(3) Although previous work (Petroni et al., 2019) has revealed that pre-trained language models (PLMs) such as BERT could learn rich factual knowledge from the pre-training on large-scale corpora, our experiments indicate that KE is still valuable for enhancing PLMs, and how to design a feasible way to combine KE and PLMs remains an exciting research direction.

Hopefully, the results of our analysis would provide some insights about how to better utilize KE for language understanding in the future.

2 Background

2.1 Knowledge Embedding

In this subsection, we first introduce several representative KE models and then summarize the KE models chosen in our experiments.

Linear Models utilize a linear combination of the relation embedding and head/tail entity embeddings to model the probability of the relational fact (Bordes et al., 2011, 2012, 2014b). LFM (Jenatton et al., 2012; Sutskever et al., 2009) is a representative linear model, which employs a relation-specific bilinear form to consider the relatedness between entities and relations. DistMult (Yang et al., 2014) further reduces the number of relation parameters in LFM via simply restricting relation matrices to be diagonal matrices, resulting in a less complicated model and better performance.

Translation Models regard the relation embedding as a translation between the head and tail entities' embeddings. Bordes et al. (2013) propose the first translation model TransE, which is simple but effective. Although TransE achieves promising results, it cannot handle the complex relations in KGs well due to its simple structure. Various translation models have been proposed to address this issue, such as TransH (Wang et al., 2014), TransR/CTransR (Lin et al., 2015), TransD (Ji et al., 2015), TranSparse (Ji et al., 2016), KG2E (He et al., 2015), and ManifoldE (Xiao et al., 2016).

Neural Models utilize neural networks to model the probability of the relational fact by taking the head/tail entity and relation embeddings as inputs. NTN (Socher et al., 2013) employs a bilinear tensor to combine two entities' embeddings via multiple aspects. Moreover, HolE (Nickel et al., 2016) uses the circular correlation of vectors to represent pairs of entities, which could combine the expressive power of the tensor product with the efficiency and simplicity of TransE. In other work, both NAM (Liu et al., 2016) and ConvE (Dettmers et al., 2018) utilize multi-layer networks to capture the interactions among entities and relations. We find that most of neural models are designed for the small-scale KG like FB15K-237. We have tried to conduct experiments on neural KE and find that the GPU memory cannot place our large-scale KG. We think the feasibility for large-scale KGs is important for KE algorithms in the application scenario, and thus the neural KE models are not included.

Complex-Valued Models exploit complex embeddings to represent the entities and relations. ComplEx (Trouillon et al., 2016) first considers complex embeddings in KE models by employing an eigenvalue decomposition model. Moreover, RotatE (Sun et al., 2019b) defines each relation as a rotation from the head entity to the tail entity in a complex vector space. Benefiting from the strong modeling ability of complex embeddings, complex-valued models achieve quite good performance compared with other KE models.

Notably, some hyperbolic models (Chami et al., 2020; Wang et al., 2020) have been proposed recently, which are mainly designed for extremely low-dimensional embeddings and cannot be easily used by neural networks. Hence, we do not choose this kind of models.

In this paper, we compare the most typical KE models from each type, including (1) Linear model: DisMult; (2) Translation model: TransE; (3) Complex-valued model: RotatE.

2.2 Utilizing External Knowledge for NLP

This subsection introduces previous work on utilizing external knowledge in different tasks, including entity-oriented and general NLP tasks.

What we call "Entity-oriented tasks" includes most of the information extraction tasks (Chang et al., 2006). These tasks naturally benefit from external knowledge about entities, and thus there are several methods using KGs for these tasks, such as entity typing (Xin et al., 2018; Liu et al., 2019a), and relation extraction (Weston et al., 2013; Han et al., 2018a; Li et al., 2019).

There are also several general NLP tasks that do not focus on entities but could effectively benefit from the information of KGs, such as question an-

swering (Bordes et al., 2014a; Miller et al., 2016; Yang and Mitchell, 2017; Huang et al., 2019; Sun et al., 2019a; Verga et al., 2020; Yasunaga et al., 2021), fact verification (Thorne et al., 2018), information retrieval (Xiong et al., 2017; Liu et al., 2018), recommendation systems (Zhang et al., 2016; Wang et al., 2018, 2019a,c; Xian et al., 2019; Wang et al., 2019b; Dhingra et al., 2020), language modeling (Ahn et al., 2016; Gu et al., 2018; Parvez et al., 2018), and dialog systems (He et al., 2017; Ghazvininejad et al., 2018). In general NLP tasks, KGs can provide external background knowledge to understand the context, such as in question answering and fact verification; or serve as external interactions between two texts for similarity measuring, such as for information retrieval and recommendation systems.

3 Knowledge-Guided Frameworks

We derive two general knowledge-guided frameworks based on previous work mentioned in the last section: knowledge augmentation framework and knowledge attention framework.

In the scenario of knowledge-driven NLU, we aim to obtain the representations of the word sequence $\boldsymbol{O}_w$ and entity sequence $\boldsymbol{O}_e$ and fuse them for prediction, where the entity sequence consists of entities appearing in text. The token sequence is denoted by $\{w_i\}_{i=1}^n$, where n is the sequence length. Meanwhile, the entity sequence is denoted by $\{e_i\}_{i=1}^m$, where m is the number of entities.

For classification tasks, such as relation classification, the representations of $\boldsymbol{O}_w$ and $\boldsymbol{O}_e$ are two vectors summarizing all information in the sequence. To use them, we input the concatenation of these two vectors into a multi-layer perceptron (MLP) to predict labels. For matching tasks, such as information retrieval, the representations of $\boldsymbol{O}_w$ and $\boldsymbol{O}_e$ are two sequences of embeddings for each word or entity. To use them, we follow the kernel method proposed by Dai et al. (2018).

Knowledge Augmentation Framework aims to directly integrate entity knowledge by treating entity sequences as external features. It could be generalized to a variety of existing work utilizing KE (Weston et al., 2013; Han et al., 2018a; Xiong et al., 2017; Liu et al., 2018). This framework formulates the entity representation $\boldsymbol{O_e}$ as

$$\boldsymbol{O_e} = \text{Enc}_e(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m), \quad (1)$$

where Enc_e is the entity encoder, which is usually an MLP, and $\boldsymbol{e}_i$ is the entity embedding of e_i. For classification tasks, Enc_e takes the concatenation of entity embeddings as input. For matching tasks, Enc_e is applied to each entity embedding.

Knowledge Attention Framework is expected to capture semantic correlations of context and entity knowledge. It utilizes entity information to gather different aspects of semantic meanings in the text sequence. This framework is also generalized to another part of knowledge-guided language understanding models (Xin et al., 2018; Kumar et al., 2018; Li et al., 2019). It treats entity embeddings as attention queries and word representations as attention key-value pairs. The process to compute the attention output $\boldsymbol{h}_{e_i}$ of entity e_i and the general representation $\boldsymbol{O_e}$ is formulated as

$$\begin{aligned} \boldsymbol{h}_{e_i} &= \boldsymbol{W}^T \text{softmax}(\boldsymbol{W}\boldsymbol{A}\boldsymbol{e}_i), \\ \boldsymbol{O_e} &= \text{Enc}_e(\boldsymbol{h}_{e_1}, \ldots, \boldsymbol{h}_{e_m}), \end{aligned} \quad (2)$$

where $\boldsymbol{A}$ is a bi-linear matrix, Enc_e is identical to that of knowledge augmentation framework, and $\boldsymbol{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$ is the word representation matrix. Note that the word representations can be contextualized, such as outputs of CNN, LSTM, or uncontextualized, such as GloVE.

4 Experimental Setup

KG Details. We adopt a sub-graph of Wikidata to train the KE models. There are $5,039,998$ entities, 927 relations, and $24,248,796$ fact triples. Note that the triples appearing in the relation classification task are removed from this KG.

Training Details. The training details of KE models and text encoders are introduced in Section A of the Appendix due to the space limitation.

Frameworks. We denote the knowledge augmentation framework by **+Aug** and the knowledge attention framework by **+Att**. We can combine the names of text encoders and frameworks to represent instantiations of these frameworks, e.g., CNN+Aug denotes the instance of CNN in the knowledge augmentation framework. We also report the results of only using KE or text in downstream tasks, which is denoted by **KE-Only** or **Text-Only**.

Evaluation Datasets. We choose four typical knowledge-driven NLU tasks, which can be divided into two types: **Entity-oriented tasks** including relation classification and entity typing, and **General NLU tasks** including information retrieval and fact verification. Examples of these tasks are shown in Figure 1. Unlike previous work designing specific models for each task, this work systematically evaluates two general knowledge-

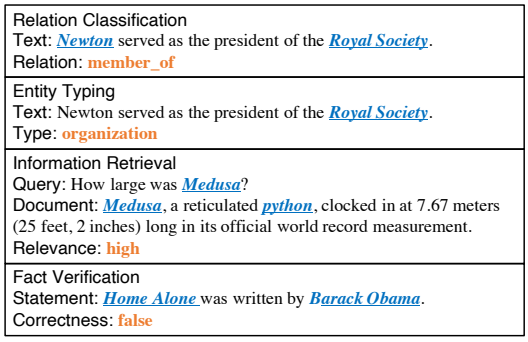

Figure 1: Examples for the evaluation tasks. The underlined mentions are the entities appearing in the inputs. The last line in each box is the corresponding label.

guided frameworks on these tasks.

(1) *Relation Classification* aims to determine the correct relation between two entities in a given sentence, which is an important task for information extraction. In this work, we choose a large-scale human-annotated relation classification dataset FewRel (Han et al., 2018b), which consists of $56,000$ instances and 80 relation classes.

(2) *Entity Typing* aims to infer the semantic type of the entity mention by its context. In this work, we adopt the large-scale entity typing dataset used by Xin et al. (2018), which contains 68 types, $860,011$ training instances, $66,860$ development instances and $68,242$ testing instances.

(3) *Information Retrieval* aims to capture the query-document relevancy by calculating the similarities between queries and documents. We use ClueWeb09 as the dataset since Xiong et al. (2017) have shown that the understanding of its many cases needs external knowledge. There are 200 queries and we adopt the five-fold cross-validation.

(4) *Fact Verification* aims to verify the correctness of a given statement regarding entities. Here, we verify the statement without evidence and keep the statements with more than two entities in FEVER (Thorne et al., 2018) to evaluate the help of KE. There are $17,918$ instances for training, $2,238$ instances for development and testing, respectively.

We select these tasks for two reasons. First, there have been many works and datasets in these tasks for knowledge integration, making the comprehensive comparison available. Second, these four tasks are representative: entity typing focuses on a single entity; relation classification focuses on the relation between two entities; information retrieval focuses on the similarities between entities; and fact verification focuses on the reasoning among entities. Notably, We exclude language modeling and dialog system because we focus on NLU tasks here.

KE	MRR	HITS@1	HITS@3	HITS@10
DistMult	0.226	0.173	0.252	0.327
TransE	0.279	0.196	0.334	0.416
RotatE	**0.302**	**0.234**	**0.345**	**0.418**

Table 1: Performance on knowledge graph completion.

Evaluation Metrics. For relation classification and fact verification, which are multi-class classification tasks, we report the prediction accuracy. For entity typing, which is a multi-label classification task, we adopt micro averaged metrics to measure the model performance. For information retrieval, which is a ranking task, we adopt precision@20 (P@20) and NDCG@20 as evaluation metrics[2].

5 Experimental Results

5.1 Effects of KE Models

We first investigate **whether KE models can help language understanding**. To this end, we evaluate the performance of different KE models on KGC and the effects of these KE models with different text encoders and knowledge-guided frameworks.

The performance on KGC is shown in Table 1. From the table, we observe that RotatE achieves the best results on all evaluation metrics.

For downstream NLU tasks, we report the results of KE-Only, Text-Only and two knowledge-guided frameworks based on three text encoders in Table 2. The best performance of each text encoder is in boldface. From the table, we find that:

(1) For CNN and LSTM, both knowledge augmentation and knowledge attention frameworks achieve better results compared to the Text-Only models on almost every task. It shows the generality and effectiveness of two knowledge-guided frameworks and the usefulness of KE models for downstream NLU tasks. Besides, knowledge augmentation works better than knowledge attention for three text encoders in most of the tasks. This suggests that directly using entity embeddings as features is more suitable for integrating KE's information into conventional text encoders.

(2) Good performance on KGC does not correlate with good performance on NLU tasks. On the one hand, RotatE, which achieves the best results in KGC, does not have consistent superior performance when applied to these NLU tasks. On the

[2]The evaluation toolkit provided by TREC (Van Gysel and de Rijke, 2018) is used.

Text Enc.	Framework	KE	RC Acc	ET P	ET R	ET F1	IR P@20	IR NDCG@20	FV Acc
-	KE-Only	DistMult	0.724	**0.738**	0.695	**0.716**	0.167	0.172	0.564
		TransE	**0.803**	0.649	**0.741**	0.692	0.157	0.165	0.576
		RotatE	0.683	0.364	0.704	0.480	**0.168**	**0.180**	**0.580**
CNN	Text-Only	-	0.668	0.768	0.626	0.690	0.258	0.276	0.737
	+Aug	DistMult	0.772	0.811	0.714	0.759	0.243	0.283	0.740
		TransE	**0.857**	**0.828**	0.733	**0.778**	**0.282**	**0.328**	0.740
		RotatE	0.796	0.812	0.675	0.738	0.271	0.320	0.743
	+Att	DistMult	0.670	0.783	0.685	0.731	0.268	0.320	0.752
		TransE	0.722	0.806	**0.737**	0.770	0.280	0.326	0.747
		RotatE	0.673	0.797	0.721	0.757	0.276	0.317	**0.754**
LSTM	Text-Only	-	0.619	0.754	0.668	0.708	0.228	0.241	0.733
	+Aug	DistMult	0.753	0.797	0.714	0.753	0.228	0.272	0.723
		TransE	**0.848**	**0.830**	0.720	**0.771**	**0.274**	0.322	0.736
		RotatE	0.774	0.780	0.703	0.740	0.267	**0.326**	0.732
	+Att	DistMult	0.645	0.795	0.722	0.757	0.235	0.255	**0.750**
		TransE	0.660	0.809	**0.735**	0.770	0.260	0.301	0.747
		RotatE	0.610	0.781	0.699	0.737	0.237	0.274	0.748
BERT	Text-Only	-	0.849	0.769	0.755	0.762	0.294	0.332	0.831
	+Aug	DistMult	0.858	0.767	**0.758**	0.762	0.295	0.330	0.831
		TransE	**0.859**	0.764	0.751	0.758	**0.307**	**0.348**	**0.832**
		RotatE	0.858	**0.787**	0.747	**0.766**	0.296	0.333	0.826
	+Att	DistMult	0.843	0.748	0.749	0.748	0.288	0.321	0.827
		TransE	0.845	0.745	0.731	0.738	0.280	0.326	0.826
		RotatE	0.845	0.763	0.743	0.753	0.297	0.338	0.831

Table 2: Performance on four NLU tasks with different KE models and text encoders. RC: Relation Classification; ET: Entity Typing; IR: Information Retrieval; FV: Fact Verification.

other hand, the performance of KE-Only is also inconsistent with two knowledge-guided frameworks. For example, in entity typing, DistMult performs best in KE-Only but TransE achieves the best result when applied in knowledge-guided frameworks. These observations indicate that the knowledge-driven frameworks may not be able to utilize the information of KE well.

(3) For fact verification, there is a tiny difference between the performance of knowledge-guided models and Text-Only models. It suggests that only using entity embeddings could not benefit this task. We will further study this phenomenon in Section 5.4 to discuss how to combine language and knowledge information.

To further investigate the performance mismatch between KGC and downstream tasks, we compare the distance of the semantic spaces between each KE model and GloVE (GloVE is the word embedding used by CNN and LSTM). We suppose that for a model using two sources (GloVE and KE), if two sources are closer to each other, the model will use them easier. Specifically, for each matched entity e, which has both word embedding and knowledge embedding, we use a unified linear matrix $\mathbf{M}$ to transform its entity embedding $\mathbf{e}$ into its corresponding word embedding $\mathbf{w}_e$ in GloVE, and define the semantic distance of e as $||\mathbf{eM} - \mathbf{w}_e||_2$. The semantic space distance between KE and GloVE is defined as the average distance of all matched entities. We show the semantic space distance of three KE models and visualize the embeddings of 40 entities with t-SNE (Maaten and Hinton, 2008) in Figure 2. From this figure, we can see that: (1) The semantic space distances of three KE models are more consistent with their performance on NLU tasks than their performance on KGC. (2) DistMult has the largest semantic space distance with GloVE, which may be one reason for its large performance gap between the KE-only framework and two knowledge-guided frameworks. According to this observation, to better utilize KE in NLU tasks, a feasible solution is to build connections between KE and text representation by joint training or designing specific fusion architectures. We will discuss this more in Section 5.4.

5.2 Analysis on KE's Helpful Information

Based on the promising results of KE models in Table 2, we further raise a question: **What informa-**

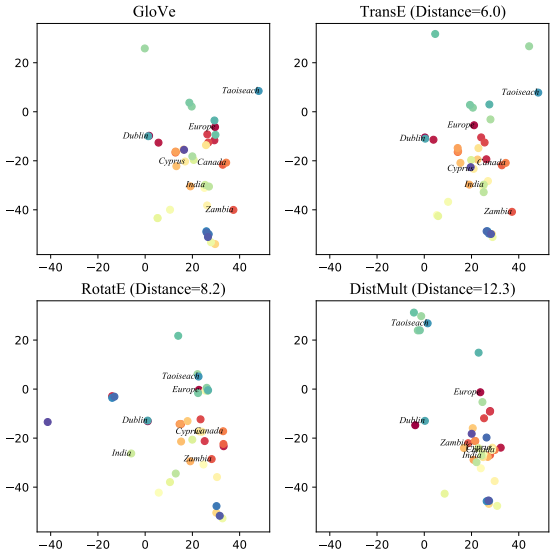

Figure 2: Visualization of GloVE and KE models. Distance: the semantic space distance. An entity in different plots has the same color. The KE with a smaller semantic distance with GloVE will plot more similarly.

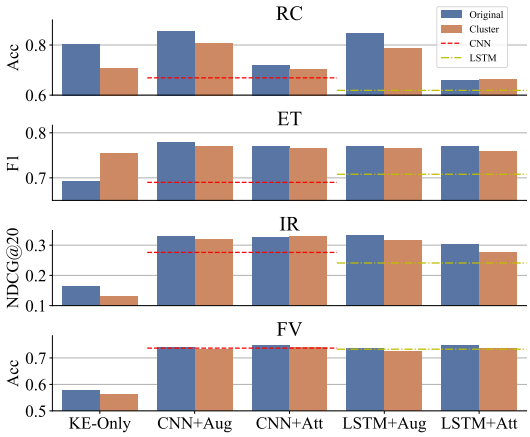

Figure 3: Comparisons of the models using entity embeddings (Origin) and cluster embeddings (Cluster).

Text Enc.	Framework	Top 1	Top 5	NDCG@20
-	KE-Only	12.71	12.95	0.165
CNN	Text-Only	11.42	11.54	0.276
	+Aug	**13.50**	**13.30**	**0.328**
	+Att	12.10	12.80	0.326
LSTM	Text-Only	11.79	12.29	0.241
	+Aug	**13.16**	**12.63**	**0.322**
	+Att	10.17	11.28	0.301
Ground Truth		13.33		-

Table 3: Average entity similarities of the query-document pairs having high relevance scores. Groud Truth is the entity similarities of ground truth pairs.

tion of KE could help language understanding? This will help us figure out the possible directions to improve the utilization of KE. Due to the space limit, we only report the results using TransE, while the conclusion of our analysis is consistent among all KE models. For the results of DistMult and RotatE, please refer to Section B of the Appendix. From our study, there are two main kinds of information in KE benefiting language understanding:

Entity Similarity Information. Intuitively, the similarities between different entities are the most important information provided by entity embeddings. Based on the similarities, we can cluster similar entities together, which could be beneficial for entity typing, and directly using the similarities could benefit information retrieval.

For entity clustering, we cluster entities with K-means (MacQueen et al., 1967), and assume the entities in the same cluster share the same information. To evaluate the effect of this information, we replace input entity embeddings with their corresponding cluster embeddings (the average of all entity embeddings in the cluster). Here, we set the number of clusters as 100. From the results in Figure 3, we can see that:

(1) In the entity typing (ET) task, the knowledge-guided frameworks using cluster embeddings perform very closely to those with original entity embeddings and the KE-Only model using cluster embeddings even achieves better performance than the KE-Only model with original embeddings. It reveals that after removing the other information from inputs, the models may further make full use of the cluster information, which is related to the entity type and entity typing mainly benefits from the cluster information of KE.

(2) For the other tasks, using cluster embeddings can also bring improvements over Text-Only (dashed lines), while there is a performance degradation compared to the models using entity embeddings. It indicates the cluster information is useful for these tasks, but there still exists other information of KE that could help these tasks.

Directly using entity similarities may play an important role in information retrieva, which emphasizes capturing the similarities between queries and documents. To verify this, we calculate the entity similarities of the top-5 query-document pairs retrieved by the models. Specifically, given a query-document pair, we calculate the cosine similarities of all entity pairs between the query and document, and average them out as the entity similarities. We report the average results of query-document pairs in Table 3. We observe that the entity similarities of the ground truth are higher than those of most

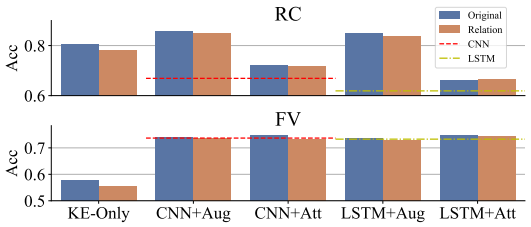

Figure 4: Comparisons of the models using entity embeddings (Origin) and relation embeddings (Relation).

models. The knowledge augmentation framework, having the highest entity similarities among models, achieves the best performance. It indicates that entity similarity is useful for information retrieval. However, we also need to consider both texts and entities' information because KE-Only has high entity similarities but the worst performance.

Entity Relation Information. Since KE is learned from relational data, the relation information of KE should be important when utilizing KE. For example, relation classification (RC) and fact verification (FV) require modeling the relations among entities in text. To extract the relation information of KE, we calculate the relation embeddings according to the scoring function of KE with the entity embeddings (e.g., the relation embedding in TransE is the difference between head and tail entity embeddings). We replace the entity embeddings with the corresponding relation embeddings in the input. The results on relation classification and fact verification are reported in Figure 4. Figure 3 and 4 show that the relation information of KE is more useful than the cluster information for relation classification. However, for fact verification, the benefit of the relation information is similar to that of the cluster information. The reason is that fact verification requires a more complex utilization of the information, which will be further discussed in Section 5.4.

5.3 Utilizing KE for PLMs

From Table 2, we notice that BERT, which is a representative pre-trained language model (PLM) having powerful representation ability, benefits little from the KE models, and is even slightly degraded in fact verification. The reason is perhaps that PLMs such as BERT have learned rich factual knowledge through pre-training from large-scale corpora (Petroni et al., 2019). Hence, we consider a question: **Could KE still benefit PLMs in language understanding?** In other words, we explore how to effectively inject KE into PLMs.

Text Enc.	Framework	RC Acc	ET F1	IR NDCG	FV Acc
RoBERTa	Text-Only	**0.852**	0.765	**0.350**	0.841
	+Aug	0.845	**0.768**	0.329	**0.842**
	+Att	0.842	0.764	0.326	0.838
KEPLER	Text-Only	**0.851**	0.767	**0.344**	**0.841**
	+Aug	0.845	0.772	0.342	0.840
	+Att	0.848	**0.780**	0.339	0.836
BERT	Text-Only	0.849	0.762	0.332	0.831
	+Aug	0.859	0.758	**0.348**	0.832
	+Att	0.845	0.738	0.326	0.826
	ERNIE	**0.878**	**0.799**	0.340	**0.842**

Table 4: The results of PLMs with different frameworks on downstream tasks.

Firstly, we evaluate whether the pre-training task will influence the ability to utilize KE. Hence, we choose RoBERTa (Liu et al., 2019b), which adopts a better pre-training paradigm than BERT, and KEPLER (Wang et al., 2019d), which adds a new pre-training task based on KGs to RoBERTa. Secondly, we evaluate ERNIE (Zhang et al., 2019), which injects KE into PLMs via designing specified model architectures. Note that ERNIE is based on BERT and can be treated as a new knowledge-guided framework. From the results in Table 4, we observe that: (1) Similar to BERT, RoBERTa also cannot benefit from the knowledge-guided frameworks in all tasks. Besides, for entity typing, KEPLER+Att achieves more than 1% improvement over KEPLER. Although the improvement is not consistent across different tasks, it still suggests the possibility of enhancing the ability to utilize knowledge by pre-training. (2) ERNIE works well in all tasks, which shows the effectiveness of designing specific modules for knowledge-guided PLMs.

In summary, the designs of both pre-training tasks and injection modules are promising for better utilizing KE for PLMs and they still need further research for new tasks and frameworks.

5.4 Error Analysis and Discussion

In this section, we analyze the errors of the knowledge guided frameworks to discover their weaknesses for further research. According to the observation on the error cases, we categorize the errors into three kinds: knowledge representation, knowledge selection, knowledge utilization. The detailed descriptions of these errors are shown in Table 5. We also provide some examples in Section C of the Appendix. We randomly sample 100 error cases from four downstream NLU tasks used in this work

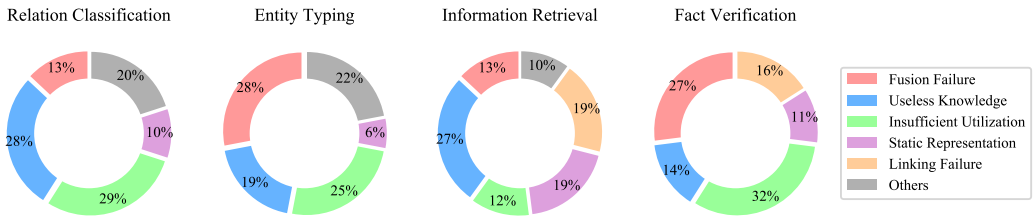

Figure 5: Types of error cases in four knowledge-driven NLU tasks.

Knowledge Representation
1. *Static Representation* - The static representations of entities, which ignore the text context, cannot satisfy the demand of tasks.
Knowledge Selection
2. *Linking Failure* - The results of entity linking contain some errors, which mislead the knowledge-guided model.
Knowledge Utilization
3. *Fusion Failure* - The KE-Only model makes a correct prediction while the fusion model does not.
4. *Useless Knowledge* - The model makes the correct prediction with text while KE causes extra noise.
5. *Insufficient Utilization* - The instances needs both text and KE information but the fusion model makes a incorrect prediction.

Table 5: Descriptions for errors. (5 types, 3 categories.)

and report the statistics of the errors in Figure 5.

From the statistics, we observe that these three kinds of errors account for a great portion of the error cases. For fact verification, where knowledge-guided frameworks do not work well, the knowledge information is still needed. Based on the results, we discuss several promising directions requiring further efforts for each kind of errors:

(1) For knowledge representation, the error of static representation appears in all four downstream tasks. Existing work (Wang et al., 2014; Zhang et al., 2015; Xu et al., 2016) have preliminarily verified the effectiveness of joint learning, which can build connections between knowledge and language. Nevertheless, how to represent knowledge based on the context is an important problem for further research, which is similar to contextualized word representation (Peters et al., 2018).

(2) For knowledge selection, linking failure appears in information retrieval and fact verification where linking results are not human-annotated. This emphasizes the importance of entity linking. Inspired by end-to-end relation extraction (Li and Ji, 2014; Miwa and Bansal, 2016), which jointly extracts entity mentions and relations, we believe entity linking can be integrated into knowledge-guided frameworks for better results. KnowBERT (Peters et al., 2019) is pioneering work, which introduces a soft entity linking mechanism.

(3) For knowledge utilization, in each task, this kind of errors accounts for more than 50% and three sub-types of error have similar portion. Although we have shown that text encoders can benefit from KE, they cannot make full use of KE and sometimes fail in knowledge fusion. What's worse, some cases indeed need external knowledge but the insufficient utilization makes it work not well. Meanwhile, directly using KE will introduce useless knowledge to the model in some cases. Hence, we need to explore how to better encode both knowledge and text information simultaneously. Recently, ERNIE (Zhang et al., 2019) and KnowBERT (Peters et al., 2019) provide us a novel perspective to fuse knowledge and language in pre-training. Besides designing novel pre-training objectives, we could also design more suitable model architectures for utilizing KE. There exist some works investigating novel model architectures to encode relational knowledge, such as memory-based models (Yang and Mitchell, 2017; Mihaylov and Frank, 2018), graph neural network-based models (Sun et al., 2018), retrieval-based models (Guu et al., 2020), etc. Nevertheless, the problem of how to effectively fuse knowledge in language understanding still remains unsolved.

6 Conclusion

In this work, we seek to better understand how KE could benefit language understanding in four knowledge-driven NLU tasks. Our comprehensive evaluation reveals (1) the performance inconsistency between knowledge graph completion and downstream NLU tasks; (2) two main kinds of useful information of KE in downstream NLU tasks; (3) how KE could benefit powerful PLMs. These observations can provide some insights for the follow-up researchers to better exploit KE in language understanding tasks.

References

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *CoRR*, abs/1608.00318.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGKDD*, pages 1247–1250.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014a. Question answering with subgraph embeddings. In *Proceedings of EMNLP*, pages 615–620.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings of AISTATS*, pages 127–135.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014b. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795.

Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*, pages 301–306.

Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014c. Open question answering with weakly supervised embedding models. In *Proceedings of ECML PKDD*, pages 165–180.

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.*, 34(4):18–42.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of ACL*, pages 6901–6914.

Chia-Hui Chang, Mohammed Kayed, Moheb R. Girgis, and Khaled F. Shaalan. 2006. A survey of web information extraction systems. *IEEE Trans. Knowl. Data Eng.*, 18(10):1411–1428.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of WSDM*, pages 126–134.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Proceedings of AAAI*, pages 1811–1818.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. 2020. Differentiable reasoning over a virtual knowledge base. In *Proceedings of ICLR*.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of AAAI*, pages 5110–5117.

Yihong Gu, Jun Yan, Hao Zhu, Zhiyuan Liu, Ruobing Xie, Maosong Sun, Fen Lin, and Leyu Lin. 2018. Language modeling with sparse product of sememe experts. In *Proceedings of EMNLP*, pages 4642–4651.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2018. Knowledge graph embedding with iterative guidance from soft rules. In *Proceedings of AAAI*, pages 4816–4823.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of ICML*.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018a. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of AAAI*, pages 4832–4839.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018b. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*, pages 4803–4809.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of ACL*, pages 1766–1776.

Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with Gaussian embedding. In *Proceedings of CIKM*, pages 623–632.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL-HLT*, pages 541–550.

Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. 2009. Understanding user's query intent with Wikipedia. In *Proceedings of the WWW*, pages 471–480.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of WSDM*, page 105–113.

Rodolphe Jenatton, Nicolas L Roux, Antoine Bordes, and Guillaume R Obozinski. 2012. A latent factor model for highly multi-relational data. In *Proceedings of NIPS*, pages 3167–3175.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*, pages 687–696.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of AAAI*, pages 5997–6004.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Abhishek Kumar, Daisuke Kawahara, and Sadao Kurohashi. 2018. Knowledge-enriched two-layered attention network for sentiment analysis. In *Proceedings of NAACL*, pages 253–258.

Pengfei Li, Kezhi Mao, Xuefeng Yang, and Qi Li. 2019. Improving relation extraction with knowledge-attention. In *Proceedings of EMNLP*, pages 229–239.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of ACL*, pages 402–412.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, pages 2181–2187.

Angli Liu, Jingfei Du, and Veselin Stoyanov. 2019a. Knowledge-augmented language model and its application to unsupervised named-entity recognition. In *Proceedings of NAACL-HLT*, pages 1142–1150.

Quan Liu, Hui Jiang, Andrew Evdokimov, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, and Yu Hu. 2016. Probabilistic reasoning via deep learning: Neural association models. *arXiv preprint arXiv:1603.07704*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-Duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *Proceedings of ACL*, pages 2395–2405.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of BSMSP*, pages 281–297.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of ACL*, pages 821–832.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of EMNLP*, pages 1400–1409.

Pasquale Minervini, Luca Costabello, Emir Muñoz, Vít Novácek, and Pierre-Yves Vandenbussche. 2017. Regularizing knowledge graph embeddings via equivalence and inversion axioms. In *Proceedings of ECML/PKDD*, volume 10534, pages 668–683.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of ACL*, pages 1105–1116.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of AAAI*, pages 1955–1961.

Ankur Padia, Konstantinos Kalpakis, Francis Ferraro, and Tim Finin. 2019. Knowledge graph fact prediction via knowledge-enriched tensor factorization. *J. Web Semant.*, 59.

Md. Rizwan Parvez, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2018. Building language models for text with named entities. In *Proceedings of ACL*, pages 2373–2383.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*.

Matthew E. Peters, Mark Neumann, Robert L Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of EMNLP-IJCNLP*, pages 43–54.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP*, pages 2463–2473.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, pages 926–934.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge. In *Proceedings of WWW*, pages 697–706.

Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. 2019a. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of EMNLP-IJCNLP*, pages 2380–2390.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of EMNLP*, pages 4231–4242.

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

Ilya Sutskever, Joshua B Tenenbaum, and Ruslan Salakhutdinov. 2009. Modelling relational data using Bayesian clustered tensor factorization. In *Proceedings of NIPS*, pages 1821–1828.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: A large-scale dataset for fact extraction and verification. pages 809–819.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of ICML*, pages 2071–2080.

Christophe Van Gysel and Maarten de Rijke. 2018. Pytrec_eval: An extremely fast python interface to trec_eval. In *Proceedings of SIGIR*, pages 873–876.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. 2020. Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge. *CoRR*, abs/2007.00849.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of WWW*, page 1835–1844.

Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019a. Multi-task feature learning for knowledge graph enhanced recommendation. In *Proceedings of WWW*, page 2000–2010.

Shen Wang, Xiaokai Wei, Cícero Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew O. Arnold, Bing Xiang, and Philip S. Yu. 2020. H2KGAT: hierarchical hyperbolic knowledge graph attention network. In *Proceedings of EMNLP*, pages 4952–4962.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019b. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of SIGKDD*, page 950–958.

Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019c. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of AAAI*, pages 5329–5336.

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

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

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of EMNLP*, pages 1366–1371.

Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of SIGIR*, page 285–294.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. From one point to a manifold: Orbit models for knowledge graph embedding. In *Proceedings of IJCAI*, pages 1315–1321.

Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2018. Improving neural fine-grained entity typing with knowledge attention. In *Proceedings of AAAI*.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-entity duet representations for document ranking. In *Proceedings of SIGIR*, pages 763–772.

Jiacheng Xu, Kan Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Knowledge graph representation with jointly structural and textual encoding. In *Proceedings of IJCAI*, pages 1318–1324.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of ACL*, pages 1436–1446.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. *CoRR*, abs/2104.06378.

Dongxu Zhang, Bin Yuan, Dong Wang, and Rong Liu. 2015. Joint semantic relevance learning with text data and graph knowledge. In *Proceedings of ACL-IJCNLP workshop*, pages 32–40.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of SIGKDD*, pages 353–362.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of ACL*, pages 1441–1451.

Zhaocheng Zhu, Shizhen Xu, Jian Tang, and Meng Qu. 2019. Graphvite: A high-performance cpu-gpu hybrid system for node embedding. In *Proceedings of WWW*, pages 2494–2504.

A Experimental Setup

Training Details of KE models. The sub-graph of Wikidata [3] is extracted by (Zhang et al., 2019). We divide these fact triples into two parts: $24,247,796$ triples for training and $1,000$ triples for validation.

In this work, we evaluate three typical KE models: DistMult, TransE, and RotatE. We use GraphVite (Zhu et al., 2019), a high-performance KE system, to train these models. We follow most of hyper-parameters provided by GraphVite for large-scale KE and only search for the best learning rate based on the result of validation, 0.6 from $\{0.2, 0.4, 0.6, 0.8\}$ for DistMult, 0.008 from $\{0.004, 0.008, 0.01, 0.02\}$ for TransE, 0.01 from $\{0.008, 0.01, 0.02, 0.04\}$ for RotatE. We set the dimension of KE as 128, which achieves the best performance on downstream NLP tasks.

Training Details of Text Encoders. In this work, we evaluate three typical text encoders. All models are optimized by Adam (Kingma and Ba, 2014) except for CNN and LSTM in information retrieval, which use SGD. The hyperparameters for these models are as follows: (1) **CNN.** We adopt a single layer CNN. The hidden size of CNN is 100. We set the batchsize as 32 for relation classification and 100 for the others. We train the models with the learning rate of 0.001 for Adam and 0.1 for SGD. The input word embeddings are GloVe (Pennington et al., 2014) with the dimension of 50. (2) **LSTM.** We adopt a single layer bi-directional LSTM. The hyper-parameters of LSTM are as the same as those of CNN. (3) **BERT.** We use $\text{BERT}_{\text{BASE}}$ released by Google and follow most of hyper-parameters provided by (Devlin et al., 2019) except that the training epoch, which varies in different tasks (10 for relation classification, 3 for entity typing, 2 for information retrieval, 30 for fact verification). The learning rate is searched from {2e-5,3e-5,5e-5}. The hyper-parameters of RoBERTa and KEPLER

[3] https://www.wikidata.org

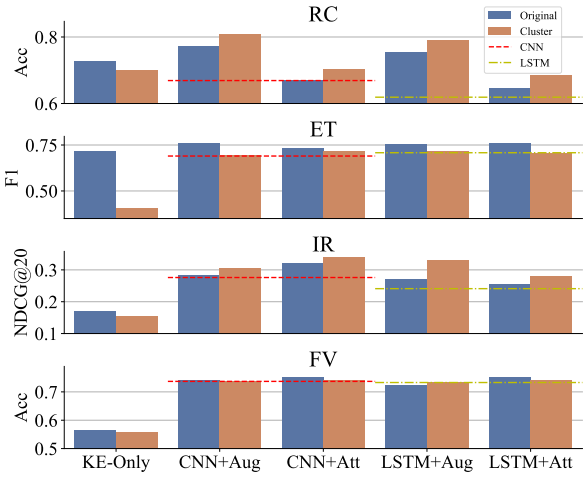

Figure 6: Comparisons of the models using entity embeddings (Origin) and cluster embeddings (Cluster). The KE model is DistMult.

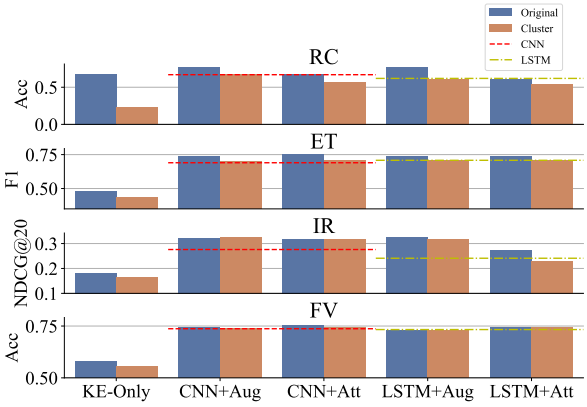

Figure 7: Comparisons of the models using entity embeddings (Origin) and cluster embeddings (Cluster). The KE model is RotatE.

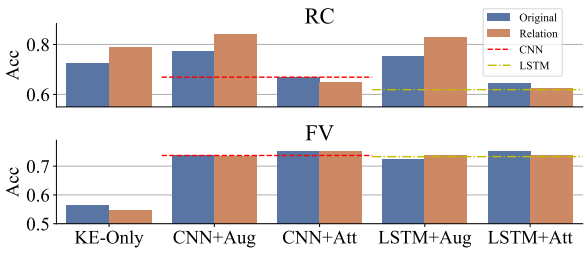

Figure 8: Comparisons between the models using the original entity embeddings and the models using the relation information of DistMult.

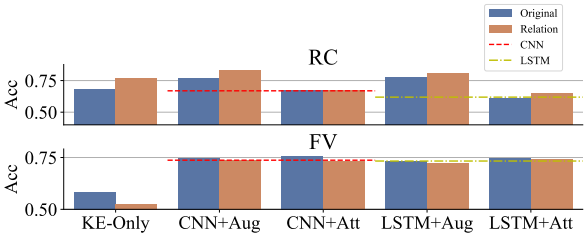

Figure 9: Comparisons between the models using the original entity embeddings and the models using the relation information of RotatE.

also follow the setting of BERT.

Computing Infrastructure and Runtime. We use NVIDIA RTX 2080Ti GPUs and each experiment uses one GPU. The average runtime of LSTM and CNN varies from several minutes to several tens of minutes according to different tasks. And the average runtime of BERT varies from several tens of minutes to several hours.

Entity Linking. For information retrieval and fact verification, which only provide the raw texts, we use TAGME [4] to link the entities mentioned in text to KGs. Meanwhile, we use the entity linking provided by the datasets for RE and ET. To avoid information leakage, we exclude the triples in the test set of RE from the KG in the training of KE.

B Analysis of KE

The results about cluster information for DistMult and RotatE are shown in Figure 6 and 7 respectively. The results about entity similarity information for DistMult and RotatE are shown in Table 6 and 7 respectively. The results about relation information for DistMult and RotatE are shown in Figure 8 and 9 respectively.

[4] https://tagme.d4science.org/tagme/

Text Enc.	Framework	Top 1	Top 5	NDCG@20
-	KE-Only	13.10	12.40	0.172
CNN	Text-Only	9.56	9.94	0.276
	+Aug	**12.77**	**12.86**	0.283
	+Att	9.99	10.53	**0.320**
LSTM	Text-Only	10.51	10.60	0.241
	+Aug	**11.6**	**11.7**	**0.272**
	+Att	10.07	10.45	0.255
Groud Truth		11.92		

Table 6: The entity similarity of the query-document pairs having high relevance score for these models. Groud Truth is the entity similarity of the ground truth pairs. The KE model is DistMult.

Text Enc.	Framework	Top 1	Top 5	NDCG@20
-	KE-Only	14.60	13.30	0.180
CNN	Text-Only	8.25	8.53	0.276
	+Aug	**9.88**	9.55	0.301
	+Att	9.23	**9.86**	**0.322**
LSTM	Text-Only	8.79	9.12	0.241
	+Aug	**11.3**	**10.8**	**0.326**
	+Att	9.54	8.89	0.274
Groud Truth		10.30		

Table 7: The entity similarity of the query-document pairs having high relevance score for these models. Groud Truth is the entity similarity of the ground truth pairs. The KE model is RotatE.

C Error Cases

We provide some example of error cases for each task.

Type of Error	Text	Label	Prediction	KE Prediction	Text Prediction
Static Representation	***Lin Liheng*** is the daughter of Lin Biao and ***Ye Qun***, nicknamed "Dou Dou".	mother	sibling	spouse	spouse
Fusion Failure	The company 's first completed game was "***Odin Sphere***" for the PlayStation 2 , which was published by ***Atlus*** .	publisher	developer	publisher	developer
Useless Knowledge	His next two films "Kutty" and "***Uthama Puthiran***", were both collaborations with director ***Mithran Jawahar***.	director	screenwriter	screenwriter	director
Inefficient Utilization	***Alphonse John Smith*** was a 20th-century bishop in the ***Catholic Church*** in the United States.	religion	main_subject	participant	language_of_work

Figure 10: Error cases of relation classification.

Type of Error	Text	Label	Prediction	KE Prediction	Text Prediction
Static Representation	The song begins as an ***acoustic guitar*** driven pop song and then shifts into a slower bridge section.	art; genre;	instrument	instrument	instrument
Fusion Failure	VET studies are offered Xavier is one of only fifteen schools in Victoria to offer ***Latin***.	language	art	language	art
Useless Knowledge	Denis Smith was born in ***Meir Stoke*** on Trent the second youngest of seven siblings.	citytown; administrative_region	sports_team	administrative_region	citytown; administrative_region
Inefficient Utilization	On 9 January 2012 Donadoni was unveiled as head coach of ***Serie A*** club Parma replacing Franco Colomba.	organization; sports_league	organization	art	profession

Figure 11: Error cases of entity typing.

Type of Error	Text	Label	Prediction	KE Prediction	Text Prediction
Static Representation	Query: ***Idaho*** state flower Document: List of ***U.S.*** state flowers – Wikipedia …	high relevance	low relevance	low relevance	low relevance
Linking Failure	Query: Dangers of ***asbestos*** Document: … ***South Dakota*** … ***Idaho*** … ***South Carolina*** … ***Hawaii*** … asbestos (missing) removal should only be performed by qualified professionals since the risks associate with an improperly conducted asbestos (missing) removal are quite high …	high relevance	low relevance	low relevance	low relevance
Fusion Failure	Query: ***Poker tournaments*** Document: Free ***Poker Tournaments*** - Free ***Poker Tournaments*** Freerolls play in a free poker tournament...	high relevance	low relevanc	high relevance	low relevanc
Useless Knowledge	Query: ***Website design*** hosting Document: Taos Web Design , Taos ***Website Design*** , Taos Web site Design , Taos Web Hosting home news about us web design cms website seo blog web hosting ecommerce ...	high relevance	low relevance	low relevance	high relevance
Inefficient Utilization	Query: ***Mothers day*** songs Document: Children 's ***Lullabies*** : ... lullabies that start with a all the pretty little horses all the pretty little ponies all the world loves to hear mothers sing all through the night version 1 ...	low relevance	high relevance	high relevance	high relevance

Figure 12: Error cases of information retrieval.

Type of Error	Text	Label	Prediction	KE Prediction	Text Prediction
Static Representation	***Connie Britton*** played a role in the first season of ***American Crime Story***.	false	true	true	true
Linking Failure	***Hansel*** and ***Gretel*** is of Mexican (missing) origin.	false	true	true	true
Fusion Failure	The ***New York Knicks*** compete in the ***National Basketball Association*** (NBA).	true	false	true	false
Useless Knowledge	***Live Through This*** has sold over 1.6 million copies in the ***United States***.	true	false	false	true
Inefficient Utilization	***Theodore Roosevelt*** attended ***Harvard College*** in 1824.	false	true	true	true

Figure 13: Error cases of fact verification.