KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation

Xiaozhi Wang[1], **Tianyu Gao**[3], **Zhaocheng Zhu**[4,5], **Zhengyan Zhang**[1]
Zhiyuan Liu[1,2*], **Juanzi Li**[1,2], **and Jian Tang**[4,6,7*]

[1]Department of CST, BNRist; [2]KIRC, Institute for AI, Tsinghua University, Beijing, China
{wangxz20,zy-z19}@mails.tsinghua.edu.cn
{liuzy,lijuanzi}@tsinghua.edu.cn
[3]Department of Computer Science, Princeton University, Princeton, NJ, USA
tianyug@princeton.edu
[4]Mila - Québec AI Institute; [5]Univesité de Montréal; [6]HEC, Montréal, Canada
zhaocheng.zhu@umontreal.ca, jian.tang@hec.ca
[7]CIFAR AI Research Chair

Abstract

Pre-trained language representation models (PLMs) cannot well capture factual knowledge from text. In contrast, knowledge embedding (KE) methods can effectively represent the relational facts in knowledge graphs (KGs) with informative entity embeddings, but conventional KE models cannot take full advantage of the abundant textual information. In this paper, we propose a unified model for **K**nowledge **E**mbedding and **P**re-trained **L**anguag**E** **R**epresentation (**KEPLER**), which can not only better integrate factual knowledge into PLMs but also produce effective text-enhanced KE with the strong PLMs. In KEPLER, we encode textual entity descriptions with a PLM as their embeddings, and then jointly optimize the KE and language modeling objectives. Experimental results show that KEPLER achieves state-of-the-art performances on various NLP tasks, and also works remarkably well as an inductive KE model on KG link prediction. Furthermore, for pre-training and evaluating KEPLER, we construct Wikidata5M[1], a large-scale KG dataset with aligned entity descriptions, and benchmark state-of-the-art KE methods on it. It shall serve as a new KE benchmark and facilitate the research on large KG, inductive KE, and KG with text. The source code can be obtained from https://github.com/THU-KEG/KEPLER.

*Correspondence to: Z. Liu and J. Tang.

[1]https://deepgraphlearning.github.io/project/wikidata5m.

1 Introduction

Recent pre-trained language representation models (PLMs) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019c) learn effective language representation from large-scale unstructured corpora with language modeling objectives and have achieved superior performances on various natural language processing (NLP) tasks. Existing PLMs learn useful linguistic knowledge from unlabeled text (Liu et al., 2019a), but they generally cannot capture the world facts well, which are typically sparse and have complex forms in text (Petroni et al., 2019; Logan et al., 2019).

By contrast, knowledge graphs (KGs) contain extensive structural facts, and knowledge embedding (KE) methods (Bordes et al., 2013; Yang et al., 2015; Sun et al., 2019) can effectively embed them into continuous entity and relation embeddings. These embeddings can not only help with the KG completion but also benefit various NLP applications (Yang and Mitchell, 2017; Zaremoodi et al., 2018; Han et al., 2018a). As shown in Figure 1, textual entity descriptions contain abundant information. Intuitively, KE methods can provide factual knowledge for PLMs, while the informative text data can also benefit KE.

Inspired by Xie et al. (2016), we use entity descriptions to bridge the gap between KE and PLM, and align the semantic space of text to the symbol space of KGs (Logeswaran et al., 2019). We propose **KEPLER**, a unified model for **K**nowledge **E**mbedding and **P**re-trained **L**anguag**E** **R**epresentation. We encode the texts

Transactions of the Association for Computational Linguistics, vol. 9, pp. 176–194, 2021. https://doi.org/10.1162/tacl_a_00360
Action Editor: Doug Downley. Submission batch: 7/2020; Revision batch: 10/2020; Published 3/2021.

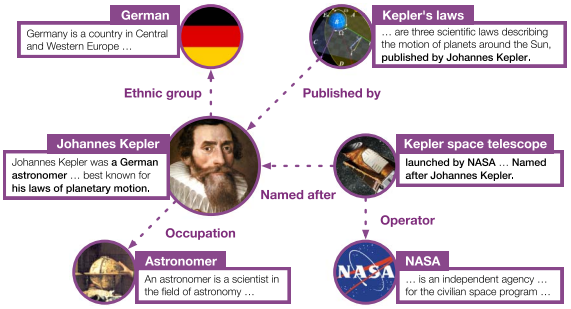

Figure 1: An example of a KG with entity descriptions. The figure suggests that descriptions contain abundant information about entities and can help to represent the relational facts between them.

and entities into a unified semantic space with the same PLM as the encoder, and jointly optimize the KE and the masked language modeling (MLM) objectives. For the KE objective, we encode the entity descriptions as entity embeddings and then train them in the same way as conventional KE methods. For the MLM objective, we follow the approach of existing PLMs (Devlin et al., 2019; Liu et al., 2019c). KEPLER has the following strengths:

As a PLM, (1) KEPLER is able to integrate factual knowledge into language representation with the supervision from KG by the KE objective. (2) KEPLER inherits the strong ability of language understanding from PLMs by the MLM objective. (3) The KE objective enhances the ability of KEPLER to extract knowledge from text since it requires the model to encode the entities from their corresponding descriptions. (4) KEPLER can be directly adopted in a wide range of NLP tasks without additional inference overhead compared to conventional PLMs since we just add new training objectives without modifying model structures.

There are also some recent works (Zhang et al., 2019; Peters et al., 2019; Liu et al., 2020) directly integrating fixed entity embeddings into PLMs to provide external knowledge. However, (1) their entity embeddings are learned by a separate KE model, and thus cannot be easily aligned with the language representation space. (2) They require an entity linker to link the text to the corresponding entities, making them suffer from the error propagation problem. (3) Compared to vanilla PLMs, their sophisticated mechanisms to link and use entity embeddings lead to additional inference overhead.

As a KE model, (1) KEPLER can take full advantage of the abundant information from entity descriptions with the help of the MLM objective. (2) KEPLER is capable of performing KE in the inductive setting, that is, it can produce embeddings for unseen entities from their descriptions, while conventional KE methods are inherently transductive and they can only learn representations for the shown entities during training. Inductive KE is essential for many real-world applications, such as updating KGs with emerging entities and KG construction, and thus is worth more investigation.

For pre-training and evaluating KEPLER, we need a KG with (1) large amounts of knowledge facts, (2) aligned entity descriptions, and (3) reasonable inductive-setting data split, which cannot be satisfied by existing KE benchmarks. Therefore, we construct Wikidata5M, containing about 5M entities, 20M triplets, and aligned entity descriptions from Wikipedia. To the best of our knowledge, it is the largest general-domain KG dataset. We also benchmark several classical KE methods and give data splits for both the transductive and the inductive settings to facilitate future research.

To summarize, our contribution is three-fold: (1) We propose KEPLER, a knowledge-enhanced PLM by jointly optimizing the KE and MLM objectives, which brings great improvements on a wide range of NLP tasks. (2) By encoding text descriptions as entity embeddings, KEPLER shows its effectiveness as a KE model, especially in the inductive setting. (3) We also introduce Wikidata5M, a new large-scale KG dataset, which shall promote the research on large-scale KG, inductive KE, and the interactions between KG and NLP.

2 KEPLER

As shown in Figure 2, KEPLER implicitly incorporates factual knowledge into language representations by jointly training with two objectives. In this section, we detailedly introduce the encoder structure, the KE and MLM objectives, and how we combine the two as a unified model.

2.1 Encoder

For the text encoder, we use Transformer architecture (Vaswani et al., 2017) in the same way

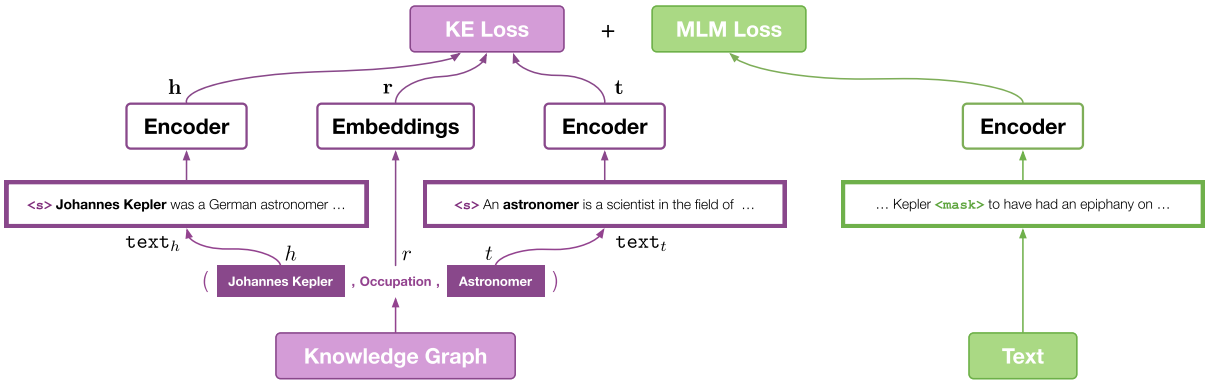

Figure 2: The KEPLER framework. We encode entity descriptions as entity embeddings and jointly train the knowledge embedding (KE) and masked language modeling (MLM) objectives on the same PLM.

as Devlin et al. (2019) and Liu et al. (2019c). The encoder takes a sequence of N tokens $(x_1, ..., x_N)$ as inputs, and computes L layers of d-dimensional contextualized representations $\mathbf{H}_i \in \mathbb{R}^{N \times d}, 1 \leq i \leq L$. Each layer of the encoder $\mathtt{E}_i$ is a combination of a multihead self-attention network and a multilayer perceptron, and the encoder gets the representation of each layer by $\mathbf{H}_i = \mathtt{E}_i(\mathbf{H}_{i-1})$. Eventually, we get a contextualized representation for each position, which could be further used in downstream tasks. Usually, there is a special token <s> added to the beginning of the text, and the output at <s> is regarded sentence representation. We denote the representation function as $\mathtt{E}_{\texttt{<s>}}(\cdot)$.

The encoder requires a tokenizer to convert plain texts into sequences of tokens. Here we use the same tokenization as RoBERTa: the Byte-Pair Encoding (BPE) (Sennrich et al., 2016).

Unlike previous knowledge-enhanced PLM works (Zhang et al., 2019; Peters et al., 2019), we do not modify the Transformer encoder structure to add external entity linkers or knowledge-integration layers. It means that our model has no additional inference overhead compared to vanilla PLMs, and it makes applying KEPLER in downstream tasks as easy as RoBERTa.

2.2 Knowledge Embedding

To integrate factual knowledge into KEPLER, we adopt the knowledge embedding (KE) objective in our pre-training. KE encodes entities and relations in knowledge graphs (KGs) as distributed representations, which benefits lots of downstream tasks, such as link prediction and relation extraction.

We first define KGs: A KG is a graph with entities as its nodes and relations between entities as its edges. We use a triplet (h, r, t) to describe a relational fact, where h, t are the head entity and the tail entity, and r is the relation type within a pre-defined relation set $\mathcal{R}$. In conventional KE models, each entity and relation is assigned a d-dimensional vector, and a scoring function is defined for training the embeddings and predicting links.

In KEPLER, instead of using stored embeddings, we encode entities into vectors by using their corresponding text. By choosing different textual data and different KE scoring functions, we have multiple variants for the KE objective of KEPLER. In this paper, we explore three simple but effective ways: entity descriptions as embeddings, entity and relation descriptions as embeddings, and entity embeddings conditioned on relations. We leave exploring advanced KE methods as our future work.

Entity Descriptions as Embeddings For a relational triplet (h, r, t), we have:

$$\begin{aligned} \mathbf{h} &= \mathtt{E}_{\texttt{<s>}}(\texttt{text}_h), \\ \mathbf{t} &= \mathtt{E}_{\texttt{<s>}}(\texttt{text}_t), \\ \mathbf{r} &= \mathbf{T}_r, \end{aligned} \tag{1}$$

where $\texttt{text}_h$ and $\texttt{text}_t$ are the descriptions for h and t, with a special token <s> at the beginning. $\mathbf{T} \in \mathbb{R}^{|\mathcal{R}| \times d}$ is the relation embeddings and $\mathbf{h}, \mathbf{t}, \mathbf{r}$ are the embeddings for h, t, and r.

We use the loss from Sun et al. (2019) as our KE objective, which adopts negative sampling (Mikolov et al., 2013) for efficient optimization:

$$\mathcal{L}_{\texttt{KE}} = -\log\sigma(\gamma - d_r(\mathbf{h},\mathbf{t})) - \sum_{i=1}^{n}\frac{1}{n}\log\sigma(d_r(\mathbf{h}'_\mathbf{i},\mathbf{t}'_\mathbf{i}) - \gamma), \quad (2)$$

where (h'_i, r, t'_i) are negative samples, γ is the margin, σ is the sigmoid function, and d_r is the scoring function, for which we choose to follow TransE (Bordes et al., 2013) for its simplicity,

$$d_r(\mathbf{h},\mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p, \quad (3)$$

where we take the norm p as 1. The negative sampling policy is to fix the head entity and randomly sample a tail entity, and vice versa.

Entity and Relation Descriptions as Embeddings A natural extension for the last method is to encode the relation descriptions as relation embeddings as well. Formally, we have,

$$\hat{\mathbf{r}} = \texttt{E}_{\texttt{<s>}}(\texttt{text}_r), \quad (4)$$

where $\texttt{text}_r$ is the description for the relation r. Then we use $\hat{\mathbf{r}}$ to replace $\mathbf{r}$ in Equations 2 and 3.

Entity Embeddings Conditioned on Relations In this manner, we use entity embeddings conditioned on r for better KE performances. The intuition is that semantics of an entity may have multiple aspects, and different relations focus on different ones (Lin et al., 2015). So we have,

$$\mathbf{h}_r = \texttt{E}_{\texttt{<s>}}(\texttt{text}_{h,r}), \quad (5)$$

where $\texttt{text}_{h,r}$ is the concatenation of the description for the entity h and the description for the relation r, with the special token <s> at the beginning and </s> in between. Correspondingly, we use $\mathbf{h}_r$ instead of $\mathbf{h}$ for Equations 2 and 3.

2.3 Masked Language Modeling

The masked language modeling (MLM) objective is inherited from BERT and RoBERTa. During pre-training, MLM randomly selects some of the input positions, and the objective is to predict the tokens at these selected positions within a fixed dictionary.

To be more specific, MLM randomly selects 15% of input positions, among which 80% are masked with the special token <mask>, 10% are replaced by other random tokens, and the rest remain unchanged. For each selected position j, the last layer of the contextualized representation $\mathbf{H}_{L,j}$ is used for a W-way classification, where W is the size of the dictionary. At last, a cross-entropy loss $\mathcal{L}_{\texttt{MLM}}$ is calculated over these selected positions.

We initialize our model with the pre-trained checkpoint of $\text{RoBERTa}_{\texttt{BASE}}$. However, we still keep MLM as one of our objectives to avoid catastrophic forgetting (McCloskey and Cohen, 1989) while training towards the KE objective. Actually, as demonstrated in Section 5.1, only using the KE objective leads to poor results in NLP tasks.

2.4 Training Objectives

To incorporate factual knowledge and language understanding into one PLM, we design a multi-task loss as shown in Figure 2 and Equation 6,

$$\mathcal{L} = \mathcal{L}_{\texttt{KE}} + \mathcal{L}_{\texttt{MLM}}, \quad (6)$$

where $\mathcal{L}_{\texttt{KE}}$ and $\mathcal{L}_{\texttt{MLM}}$ are the losses for KE and MLM correspondingly. Jointly optimizing the two objectives can implicitly integrate knowledge from external KGs into the text encoder, while preserving the strong abilities of PLMs for syntactic and semantic understanding. Note that those two tasks only share the text encoder, and for each mini-batch, text data sampled for KE and MLM are not (necessarily) the same. This is because seeing a variety of text (instead of just entity descriptions) in MLM can help the model to have better language understanding ability.

2.5 Variants and Implementations

We introduce the variants of KEPLER and the pre-training implementations here. The fine-tuning details will be introduced in Section 4.

KEPLER Variants

We implement multiple versions of KEPLER in experiments to explore the effectiveness of our pre-training framework. We use the same denotations in Section 4 as below.

KEPLER-Wiki is the principal model in our experiments, which adopts Wikidata5M (Section 3) as the KG and the entity-description-as-embedding method (Equation 1). All other variants, if not specified, use the same settings. KEPLER-Wiki achieves the best performances on most tasks.

KEPLER-WordNet uses the WordNet (Miller, 1995) as its KG source. WordNet is an English lexical graph, where nodes are lemmas and synsets, and edges are their relations. Intuitively, incorporating WordNet can bring lexical knowledge and thus benefits NLP tasks. We use the same WordNet 3.0 as in KnowBert (Peters et al., 2019), which is extracted from the `nltk`[2] package.

KEPLER-W+W takes both Wikidata5M and WordNet as its KGs. To jointly train with two KG datasets, we modify the objective in Equation 6 as

$$\mathcal{L} = \mathcal{L}_{\texttt{Wiki}} + \mathcal{L}_{\texttt{WordNet}} + \mathcal{L}_{\texttt{MLM}}, \qquad (7)$$

where $\mathcal{L}_{\texttt{Wiki}}$ and $\mathcal{L}_{\texttt{WordNet}}$ are losses from Wikidata5M and WordNet respectively.

KEPLER-Rel uses the entity and relation descriptions as embeddings method (Equation 4). As the relation descriptions in Wikidata are short (11.7 words on average) and homogeneous, encoding relation descriptions as relation embeddings results in worse performance as shown in Section 4.

KEPLER-Cond uses the entity-embedding-conditioned-on-relation method (Equation 5). This model achieves superior results in link prediction tasks, both transductive and inductive (Section 4.3).

KEPLER-OnlyDesc trains the MLM objective directly on the entity descriptions from the KE objective rather than uses the English Wikipedia and BookCorpus as other versions of KEPLER. However, as the entity description data are smaller (2.3 GB vs 13 GB) and homogeneous, it harms the general language understanding ability and thus performs worse (Section 4.2).

KEPLER-KE only adopts the KE objective in pre-training, which is an ablated version of KEPLER-Wiki. It is used to show the necessity of the MLM objective for language understanding.

Pre-training Implementation

In practice, we choose RoBERTa (Liu et al., 2019c) as our base model and implement KEPLER in the fairseq framework (Ott et al., 2019) for pre-training. Due to the computing resource limit, we choose the `BASE` size ($L = 12$, $d = 768$) and use the released `roberta.base` parameters for initialization, which is a common practice to save pre-training time (Zhang et al., 2019; Peters et al., 2019). For the MLM objective, we use the English Wikipedia (2,500M words) and BookCorpus (800M words) (Zhu et al., 2015) as our pre-training corpora (except KEPLER-OnlyDesc). We extract text from these two corpora in the same way as Devlin et al. (2019). For the KE objective, we encode the first 512 tokens of entity descriptions from the English Wikipedia as entity embeddings.

We set the γ in Equation 2 as 4 and 9 for NLP and KE tasks respectively, and we use the models pre-trained with 10 and 30 epochs for NLP and KE. Specially, the γ is 1 for KEPLER-WordNet. The two hyperparameters are tuned by multiple trials for γ in $\{1, 2, 4, 6, 9\}$ and the number of epochs in $\{5, 10, 20, 30, 40\}$, and we select the model by performances on TACRED (F-1) and inductive link prediction (HITS@10). We use gradient accumulation to achieve a batch size of 12,288.

3 Wikidata5M

As shown in Section 2, to train KEPLER, the KG dataset should (1) be large enough, (2) contain high-quality textual descriptions for its entities and relations, and (3) have a reasonable inductive setting, which most existing KG datasets do not provide. Thus, based on Wikidata[3] and English Wikipedia,[4] we construct Wikidata5M, a large-scale KG dataset with aligned text descriptions from corresponding Wikipedia pages, and also an inductive test set. In the following sections, we first introduce the data collection (Section 3.1) and the data split (Section 3.2), and then provide the results of representative KE methods on the dataset (Section 3.3).

3.1 Data Collection

We use the July 2019 dump of Wikidata and Wikipedia. For each entity in Wikidata, we align it to its Wikipedia page and extract the first section as its description. Entities with no pages or with descriptions fewer than 5 words are discarded.

[2] `https://www.nltk.org`.

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

[4] `https://en.wikipedia.org`.

Dataset	#entity	#relation	#training	#validation	#test
FB15K	14,951	1,345	483,142	50,000	59,07
WN18	40,943	18	141,442	5,000	5,00
FB15K-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134
Wikidata5M	4,594,485	822	20,614,279	5,163	5,133

Table 1: Statistics of Wikidata5M (transductive setting) compared with existing KE benchmarks.

Entity Type	Occurrence	Percentage
Human	1,517,591	33.0%
Taxon	363,882	7.9%
Wikimedia list	118,823	2.6%
Film	114,266	2.5%
Human Settlement	110,939	2.4%
Total	2,225,501	48.4%

Table 2: Top-5 entity categories in Wikidata5M.

We retrieve all the relational facts in Wikidata. A fact is considered to be valid when both of its entities are not discarded, and its relation has a non-empty page in Wikidata. The final KG contains 4,594,485 entities, 822 relations and 20,624,575 triplets. Statistics of Wikidata5M along with four other widely used benchmarks are shown in Table 1. Top-5 entity categories are listed in Table 2. We can see that Wikidata5M is much larger than other KG datasets, covering various domains.

3.2 Data Split

For Wikidata5M, we take two different settings: the transductive setting and the inductive setting.

The **transductive setting** (shown in Table 1) is adopted in most KG datasets, where the entities are shared and the triplet sets are disjoint across training, validation and test. In this case, KE models are expected to learn effective entity embeddings only for the entities in the training set. In the **inductive setting** (shown in Table 3), the entities and triplets are mutually disjoint across training, validation and test. We randomly sample some connected subgraphs as the validation and test set. In the inductive setting, the KE models should produce embeddings for the unseen entities given side features like descriptions, neighbors, etc. The inductive setting is more challenging and also meaningful in real-world applications, where entities in KGs experience open-ended growth, and the inductive ability is crucial for online KE methods.

Subset	#entity	#relation	#triplet
Training	4,579,609	822	20,496,514
Validation	7,374	199	6,699
Test	7,475	201	6,894

Table 3: Statistics of Wikidata5M inductive setting.

Although Wikidata5M contains massive entities and triplets, our validation and test set are not large, which is limited by the standard evaluation method of link prediction (Section 3.3). Each episode of evaluation requires $|\mathcal{E}| \times |\mathcal{T}| \times 2$ times of KE score calculation, where $|\mathcal{E}|$ and $|\mathcal{T}|$ are the total number of entities and the number of triplets in test set respectively. As Wikidata5M contains massive entities, the evaluation is very time-consuming, hence we have to limit the test set to thousands of triplets to ensure tractable evaluations. This indicates that large-scale KE urges a more efficient evaluation protocol. We will leave exploring it to future work.

3.3 Benchmark

To assess the challenges of Wikidata5M, we benchmark several popular KE models on our dataset in the transductive setting (as they inherently do not support the inductive setting). Because their original implementations do not scale to Wikidata5M, we benchmark these methods with GraphVite (Zhu et al., 2019), a multi-GPU KE toolkit.

In the transductive setting, for each test triplet (h, r, t), the model ranks all the entities by scoring $(h, r, t'), t' \in \mathcal{E}$, where $\mathcal{E}$ is the entity set excluding other correct t. The evaluation metrics, MRR (mean reciprocal rank), MR (mean rank), and HITS@$\{1,3,10\}$, are based on the rank of the

Method	MR	MRR	HITS@1	HITS@3	HITS@10
TransE (Bordes et al., 2013)	109370	25.3	17.0	31.1	39.2
DistMult (Yang et al., 2015)	211030	25.3	20.8	27.8	33.4
ComplEx (Trouillon et al., 2016)	244540	28.1	22.8	31.0	37.3
SimplE (Kazemi and Poole, 2018)	115263	29.6	25.2	31.7	37.7
RotatE (Sun et al., 2019)	89459	29.0	23.4	32.2	39.0

Table 4: Performance of different KE models on Wikidata5M (% except MR).

correct tail entity t among all the entities in $\mathcal{E}$. Then we do the same thing for the head entities. We report the average results over all test triplets and over both head and tail entity predictions.

Table 4 shows the results of popular KE methods on Wikidata5M, which are all significantly lower than on existing KG datasets like FB15K-237, WN18RR, and so forth. It demonstrates that Wikidata5M is more challenging due to its large scale and high coverage. The results advocate for more efforts towards large-scale KE.

4 Experiments

In this section, we introduce the experiment settings and results of our model on various NLP and KE tasks, along with some analyses on KEPLER.

4.1 Experimental Setting

Baselines In our experiments, **RoBERTa** is an important baseline since KEPLER is based on it (all mentioned models are of BASE size if not specified). As we cannot afford the full RoBERTa corpora (126 GB, and we only use 13 GB) in KEPLER pre-training, we implement **Our RoBERTa** for direct comparisons to KEPLER. It is initialized by $\text{RoBERTa}_{\text{BASE}}$ and is further trained with the MLM objective on the same corpora as KEPLER.

We also evaluate recent knowledge-enhanced PLMs, including $\textbf{ERNIE}_{\textbf{BERT}}$ (Zhang et al., 2019) and $\textbf{KnowBert}_{\textbf{BERT}}$ (Peters et al., 2019). As ERNIE and our principal model KEPLER-Wiki only use Wikidata, we take KnowBert-Wiki in the experiments to ensure fair comparisons with the same knowledge source. Considering KEPLER is based on RoBERTa, we reproduce the two models with RoBERTa too ($\textbf{ERNIE}_{\textbf{RoBERTa}}$ and $\textbf{KnowBert}_{\textbf{RoBERTa}}$). The reproduction of KnowBert is based on its original implementation.[5]

[5] https://github.com/allenai/kb.

On relation classification, we also compare with MTB (Baldini Soares et al., 2019), which adopts ''matching the blank'' pre-training. Different from other baselines, the original MTB is based on $\text{BERT}_{\text{LARGE}}$ (denoted by $\textbf{MTB (BERT}_{\textbf{LARGE}}\textbf{)}$). For a fair comparison under the same model size, we reimplement MTB with $\text{BERT}_{\text{BASE}}$ (**MTB**).

Hyperparameter The pre-training settings are in Section 2.5. For fine-tuning on downstream tasks, we set KEPLER hyperparameters the same as reported in KnowBert on TACRED and OpenEntity. On FewRel, we set the learning rate as 2e-5 and batch size as 20 and 4 for the Proto and PAIR frameworks respectively. For GLUE, we follow the hyperparameters reported in RoBERTa. For baselines, we keep their original hyperparameters unchanged or use the best trial in KEPLER searching space if no original settings are available.

4.2 NLP Tasks

In this section, we demonstrate the performance of KEPLER and its baselines on various NLP tasks.

Relation Classification

Relation classification requires models to classify relation types between two given entities from text. We evaluate KEPLER and other baselines on two widely used benchmarks: TACRED and FewRel.

TACRED (Zhang et al., 2017) has 42 relations and 106,264 sentences. Here we follow the settings of Baldini Soares et al. (2019), where we add four special tokens before and after the two entity mentions, and concatenate the representations at the beginnings of the two entities for classification. Note that the original KnowBert also takes entity types as inputs, which is different from Zhang et al. (2019); Baldini Soares et al. (2019). To ensure fair comparisons, we re-evaluate KnowBert with the

Model	P	R	F-1
BERT	67.2	64.8	66.0
$\text{BERT}_{\text{LARGE}}$	–	–	70.1
MTB	69.7	67.9	68.8
MTB ($\text{BERT}_{\text{LARGE}}$)	–	–	71.5
$\text{ERNIE}_{\text{BERT}}$	70.0	66.1	68.0
$\text{KnowBert}_{\text{BERT}}$	**73.5**	64.1	68.5
RoBERTa	70.4	71.1	70.7
$\text{ERNIE}_{\text{RoBERTa}}$	**73.5**	68.0	70.7
$\text{KnowBert}_{\text{RoBERTa}}$	71.9	69.9	70.9
Our RoBERTa	70.8	69.6	70.2
KEPLER-Wiki	71.5	**72.5**	**72.0**
KEPLER-WordNet	71.4	71.3	71.3
KEPLER-W+W	71.1	72.0	71.5
KEPLER-Rel	71.3	70.9	71.1
KEPLER-Cond	72.1	70.7	71.4
KEPLER-OnlyDesc	72.3	69.1	70.7
KEPLER-KE	63.5	60.5	62.0

Table 5: Precision, recall, and F-1 on TACRED (%). KnowBert results are different from the original paper since different task settings are used.

same setting as other baselines, thus the reported results are different from the original paper.

From the TACRED results in Table 5, we can observe that: (1) KEPLER-Wiki is the best one among KEPLER variants and significantly outperforms all the baselines, while other versions of KEPLER also achieve good results. It demonstrates the effectiveness of KEPLER on integrating factual knowledge into PLMs. Based on the result, we use KEPLER-Wiki as the principal model in the following experiments. (2) KEPLER-WordNet shows a marginal improvement over Our RoBERTa, while KEPLER-W+W underperforms KEPLER-Wiki. It suggests that pre-training with WordNet only has limited benefits in the KEPLER framework. We will explore how to better combine different KGs in our future work.

FewRel (Han et al., 2018b) is a few-shot relation classification dataset with 100 relations and 70,000 instances, which is constructed with Wikipedia text and Wikidata facts. Furthermore, Gao et al. (2019) propose **FewRel 2.0**, adding a domain adaptation challenge with a new medical-domain test set.

FewRel takes the N-way K-shot setting. Relations in the training and test sets are disjoint. For every evaluation episode, N relations, K supporting samples for each relation, and several query sentences are sampled from the test set. The models are required to classify queries into one of the N relations only given the sampled $N \times K$ instances.

We use two state-of-the-art few-shot frameworks: **Proto** (Snell et al., 2017) and **PAIR** (Gao et al., 2019). We replace the text encoders with our baselines and KEPLER and compare the performance. Because FewRel 1.0 is constructed with Wikidata, we remove all the triplets in its test set from Wikidata5M to avoid information leakage for KEPLER. However, we cannot control the KGs used in our baselines. We mark the models utilizing Wikidata and have information leakage risk with † in Table 6.

As Table 6 shows, KEPLER-Wiki achieves the best performance over the BASE-size PLMs in most settings. From the results, we also have some interesting observations: (1) RoBERTa consistently outperforms BERT on various NLP tasks (Liu et al., 2019c), yet the RoBERTa-based models here are comparable or even worse than BERT-based models in the PAIR framework. Because PAIR uses sentence concatenation, this result may be credited to the next sentence prediction (NSP) objective of BERT. (2) KEPLER brings improvements on FewRel 2.0, while ERNIE and KnowBert even degenerate in most of the settings. It indicates that the paradigms of ERNIE and KnowBert cannot well generalize to new domains which may require much different entity linkers and entity embeddings. On the other hand, KEPLER not only learns better entity representations but also acquires a general ability to extract factual knowledge from the context across different domains. We further verify this in Section 5.5. (3) KnowBert underperforms ERNIE in FewRel while it typically achieves better results on other tasks. This may be because it uses the TuckER (Balazevic et al., 2019) KE model while ERNIE and KEPLER follow TransE (Bordes et al., 2013). We will explore the effects of different KE methods in the future.

We also have another two observations with regard to ERNIE and MTB: (1) ERNIE performs the best on 1-shot settings of FewRel 1.0. We believe this is because that the knowledge embedding injection of ERNIE has particular advantages in this case, since it directly brings

knowledge about entities. When using 5-shot (supporting text provides more information) and FewRel 2.0 (ERNIE does not have knowledge for biomedical entities), KEPLER outperforms ERNIE. (2) Though MTB ($\text{BERT}_{\text{LARGE}}$) is the state-of-the-art model on FewRel, its $\text{BERT}_{\text{BASE}}$ version does not outperform other knowledge-enhanced PLMs, which suggests that using large models contributes much to its gain. We also notice that when combined with PAIR, MTB suffers an obvious performance drop, which may be because its pre-training objective degenerates sentence-pair tasks.

Entity Typing

Entity typing requires to classify given entity mentions into pre-defined types. For this task, we carry out evaluations on OpenEntity (Choi et al., 2018) following the settings in Zhang et al. (2019). OpenEntity has 6 entity types and 2,000 instances for training, validation and test each.

To identify the entity mentions of interest, we add two special tokens before and after the entity spans, and use the representations of the first special tokens for classification. As shown in Table 7, KEPLER-Wiki achieves state-of-the-art results. Note that the KnowBert results are different from the original paper since we use KnowBert-Wiki here rather than KnowBert-W+W to ensure the same knowledge resource and fair comparisons. KEPLER does not perform linking or entity embedding pre-training like ERNIE and KnowBert, which bring them special advantages in entity span tasks. However, KEPLER still outperforms these baselines, which proves its effectiveness.

GLUE

The General Language Understanding Evaluation (GLUE) (Wang et al., 2019b) collects several natural language understanding tasks and is widely used for evaluating PLMs. In general, solving GLUE does not require factual knowledge (Zhang et al., 2019) and we use it to examine whether KEPLER harms the general language understanding ability.

Table 8 shows the GLUE results. We can observe that KEPLER-Wiki is close to Our RoBERTa, suggesting that while incorporating factual knowledge, KEPLER maintains a strong language understanding ability. However, there are significant performance drops of KEPLER-OnlyDesc, which indicates that the small-scale entity description data are not sufficient for training KEPLER with MLM.

For the small datasets STS-B, MRPC and RTE, directly fine-tuning models on them typically result in unstable performance. Hence we fine-tune models on a large-scale dataset (here we use MNLI) first and then further fine-tune them on the small datasets. The method has been shown to be effective (Wang et al., 2019a) and is also used in the original RoBERTa paper (Liu et al., 2019c).

4.3 KE Tasks

We show how KEPLER works as a KE model, and evaluate it on Wikidata5M in both the transductive link prediction setting and the inductive setting.

Experimental Settings

In link prediction, the entity and relation embeddings of KEPLER are obtained as described in Section 2.2 and 2.5. The evaluation method is described in Section 3.3. We also add RoBERTa and Our RoBERTa as baselines. They adopt Equations 1 and 4 to acquire entity and relation embeddings, and use Equation 3 as their scoring function.

In the transductive setting, we compare our models with **TransE** (Bordes et al., 2013). We set its dimension as 512, negative sampling size as 64, batch size as 2048, and learning rate as 0.001 after hyperparameter searching. The negative sampling size is crucial for the performance on KE tasks, but limited by the model complexity, KEPLER can only take a negative size of 1. For a direct comparison to intuitively show the benefits of pre-training, we set a baseline **TransE**†, which also uses 1 as the negative sampling size and keeps the other hyperparameters unchanged.

Because conventional KE methods like TransE inherently cannot provide embeddings for unseen entities, we take **DKRL** (Xie et al., 2016) as our baseline in the KE experiments, which utilizes convolutional neural networks to encode entity descriptions as embeddings. We set its dimension as 768, negative sampling size as 64, batch size as 1024, and learning rate as 0.0005.

Transductive Setting

Table 9a shows the results of the transductive setting. We observe that:

Model	FewRel 1.0				FewRel 2.0			
	5-1	5-5	10-1	10-5	5-1	5-5	10-1	10-5
MTB (BERT$_{\texttt{LARGE}}$)†	93.86	97.06	89.20	94.27	–	–	–	–
Proto (BERT)	80.68	89.60	71.48	82.89	40.12	51.50	26.45	36.93
Proto (MTB)	81.39	91.05	71.55	83.47	52.13	76.67	48.28	69.75
Proto (ERNIE$_{\texttt{BERT}}$)†	**89.43**	94.66	**84.23**	90.83	49.40	65.55	34.99	49.68
Proto (KnowBert$_{\texttt{BERT}}$)†	86.64	93.22	79.52	88.35	64.40	79.87	51.66	69.71
Proto (RoBERTa)	85.78	95.78	77.65	92.26	64.65	82.76	50.80	71.84
Proto (Our RoBERTa)	84.42	95.30	76.43	91.74	61.98	83.11	48.56	72.19
Proto (ERNIE$_{\texttt{RoBERTa}}$)†	87.76	95.62	80.14	91.47	54.43	80.48	37.97	66.26
Proto (KnowBert$_{\texttt{RoBERTa}}$)†	82.39	93.62	76.21	88.57	55.68	71.82	41.90	58.55
Proto (KEPLER-Wiki)	88.30	**95.94**	81.10	**92.67**	**66.41**	**84.02**	**51.85**	**73.60**
PAIR (BERT)	88.32	93.22	80.63	87.02	**67.41**	78.57	**54.89**	66.85
PAIR (MTB)	83.01	87.64	73.42	78.47	46.18	70.50	36.92	55.17
PAIR (ERNIE$_{\texttt{BERT}}$)†	**92.53**	94.27	**87.08**	89.13	56.18	68.97	43.40	54.35
PAIR (KnowBert$_{\texttt{BERT}}$)†	88.48	92.75	82.57	86.18	66.05	77.88	50.86	67.19
PAIR (RoBERTa)	89.32	93.70	82.49	88.43	66.78	81.84	53.99	70.85
PAIR (Our RoBERTa)	89.26	93.71	83.32	89.02	63.22	77.66	49.28	65.97
PAIR (ERNIE$_{\texttt{RoBERTa}}$)†	87.46	94.11	81.68	87.83	59.29	72.91	48.51	60.26
PAIR (KnowBert$_{\texttt{RoBERTa}}$)†	85.05	91.34	76.04	85.25	50.68	66.04	37.10	51.13
PAIR (KEPLER-Wiki)	90.31	**94.28**	85.48	**90.51**	67.23	**82.09**	54.32	**71.01**

Table 6: Accuracies (%) on the FewRel dataset. N-K indicates the N-way K-shot setting. MTB uses the LARGE size and all the other models use the BASE size. † indicates oracle models which may have seen facts in the FewRel 1.0 test set during pre-training.

Model	P	R	F-1
UFET (Choi et al., 2018)	77.4	60.6	68.0
BERT	76.4	71.0	73.6
ERNIE$_{\texttt{BERT}}$	78.4	72.9	75.6
KnowBert$_{\texttt{BERT}}$	77.9	71.2	74.4
RoBERTa	77.4	73.6	75.4
ERNIE$_{\texttt{RoBERTa}}$	80.3	70.2	74.9
KnowBert$_{\texttt{RoBERTa}}$	78.7	72.7	75.6
Our RoBERTa	75.1	73.4	74.3
KEPLER-Wiki	77.8	74.6	**76.2**

Table 7: Entity typing results on OpenEntity (%).

Model	MNLI (m/mm) 392K	QQP 363K	QNLI 104K	SST-2 67K
RoBERTa	87.5/87.2	91.9	92.7	94.8
Our RoBERTa	87.1/86.8	90.9	92.5	94.7
KEPLER-Wiki	87.2/86.5	91.7	92.4	94.5
KEPLER-OnlyDesc	85.9/85.6	90.8	92.4	94.4
Model	**CoLA 8.5K**	**STS-B 5.7K**	**MRPC 3.5K**	**RTE 2.5K**
RoBERTa	63.6	91.2	90.2	80.9
Our RoBERTa	63.4	91.1	88.4	82.3
KEPLER-Wiki	63.6	91.2	89.3	85.2
KEPLER-OnlyDesc	55.8	90.2	88.5	78.3

Table 8: GLUE results on the dev set (%). All the results are medians over 5 runs. We report F-1 scores for QQP and MRPC, Spearman correlations for STS-B, and accuracy scores for the other tasks. The ''m/mm'' stands for matched/mismatched evaluation sets for MNLI (Williams et al., 2018).

(1) KEPLER underperforms TransE. It is reasonable since KEPLER is limited by its large model size, and thus cannot use a large negative sampling size (1 for KEPLER, while typical KE methods use 64 or more) and more training epochs (30 vs. 1000 for TransE), which are crucial for KE (Zhu et al., 2019). On the other hand, KEPLER and its variants perform much better than TransE† (with a negative sampling size of 1), showing that using the same negative sampling size, KEPLER can benefit from pre-trained language representations and textual entity descriptions so that outperform TransE. In the future, we will explore reducing the model size of KEPLER to take advantage of both large negative sampling size and pre-training.

(2) The vanilla RoBERTa perform poorly in KE while KEPLER achieves favorable performances,

Model	MR	MRR	HITS@1	HITS@3	HITS@10
TransE (Bordes et al., 2013)	109370	**25.3**	17.0	**31.1**	**39.2**
TransE†	406957	6.0	1.8	8.0	13.6
DKRL (Xie et al., 2016)	31566	16.0	12.0	18.1	22.9
RoBERTa	1381597	0.1	0.0	0.1	0.3
Our RoBERTa	1756130	0.1	0.0	0.1	0.2
KEPLER-KE	76735	8.2	4.9	8.9	15.1
KEPLER-Rel	15820	6.6	3.7	7.0	11.7
KEPLER-Wiki	**14454**	15.4	10.5	17.4	24.4
KEPLER-Cond	20267	21.0	**17.3**	22.4	27.7

(a) Transductive results on Wikidata5M (% except MR). TransE† denotes a TransE modeled trained with the same negative sampling size (1) as KEPLER.

Model	MR	MRR	HITS@1	HITS@3	HITS@10
DKRL (Xie et al., 2016)	78	23.1	5.9	32.0	54.6
RoBERTa	723	7.4	0.7	1.0	19.6
Our RoBERTa	1070	5.8	1.9	6.3	13.0
KEPLER-KE	138	17.8	5.7	22.9	40.7
KEPLER-Rel	35	33.4	15.9	43.5	66.1
KEPLER-Wiki	32	35.1	15.4	46.9	71.9
KEPLER-Cond	**28**	**40.2**	**22.2**	**51.4**	**73.0**

(b) Inductive results on Wikidata5M (% except MR).

Table 9: Link prediction results on Wikidata5M transductive and inductive settings.

which demonstrates the effectiveness of our multi-task pre-training to infuse factual knowledge.

(3) Among the KEPLER variants, KEPLER-Cond has superior results, which substantiates the intuition in Section 2.2. KEPLER-Rel performs worst, which we believe is due to the short and homogeneous relation descriptions of Wikidata. KEPLER-KE significantly underperforms KEPLER-Wiki, which suggests that the MLM objective is necessary as well for the KE tasks to build effective language representation.

(4) We also notice that DKRL performs well on the transductive setting and the result is close to KEPLER. We believe this is because DKRL takes a much smaller encoder (CNN) and thus is easier to train. In the more difficult inductive setting, the gap between DKRL and KEPLER is larger, which better shows the language understanding ability of KEPLER to utilize textual entity descriptions.

Inductive Setting

Table 9b shows the Wikidata5M inductive results. KEPLER outperforms DKRL and RoBERTa by a large margin, demonstrating the effectiveness of our joint training method. But KEPLER results are still far from ideal performances required by practical applications (constructing KG from scratch, etc.), which urges further efforts on inductive KE. Comparisons among KEPLER variants are consistent with in the transductive setting.

In addition, we clarify why results in the inductive setting are much higher than the transductive setting, while the inductive setting is more difficult: As shown in Tables 1 and 3, the entities involved in the inductive evaluation is much less than the transductive setting (7,475 vs. 4,594,485). Considering the KE evaluation metrics are based on entity ranking, it is reasonable to see higher values in the inductive setting. The performance in different settings should not be directly compared.

5 Analysis

In this section, we analyze the effectiveness and efficiency of KEPLER with experiments. All the hyperparameters are the same as reported in Section 4.1, including models in the ablation study.

Model	P	R	F-1
Our RoBERTa	70.8	69.6	70.2
KEPLER-KE	63.5	60.5	62.0
KEPLER-Wiki	71.5	72.5	72.0

Table 10: Ablation study results on TACRED (%).

5.1 Ablation Study

As shown in Equation 6, KEPLER takes a multi-task loss. To demonstrate the effectiveness of the joint objective, we compare full KEPLER with models trained with only the MLM loss (**Our RoBERTa**) and only the KE loss (**KEPLER-KE**) on TACRED. As demonstrated in Table 10, compared to KEPLER-Wiki, both ablation models suffer significant drops. It suggests that the performance gain of KEPLER is credited to the joint training towards both objectives.

5.2 Knowledge Probing Experiment

Section 4.2 shows that KEPLER can achieve significant improvements on NLP tasks requiring factual knowledge. To further verify whether KEPLER can better integrate factual knowledge into PLMs and help to recall them, we conduct experiments on LAMA (Petroni et al., 2019), a widely used knowledge probe. LAMA examines PLMs' abilities on recalling relational facts by cloze-style questions. For instance, given a natural language template ''Paris is the capital of <mask>'', PLMs are required to predict the masked token without fine-tuning. LAMA reports the micro-averaged precision at one (P@1) scores. However, Poerner et al. (2020) present that LAMA contains some easy questions which can be answered with superficial clues like entity names. Hence we also evaluate the models on LAMA-UHN (Poerner et al., 2020), which filters out the questionable templates from the Google-RE and T-REx corpora of LAMA.

The evaluation results are shown in Table 11, from which we have the following observations: (1) KEPLER consistently outperforms the vanilla PLM baseline Our RoBERTa in almost all the settings except ConceptNet, which focuses on commonsense knowledge rather than factual knowledge. It indicates that KEPLER can indeed better integrate factual knowledge. (2) Although KEPLER-W+W cannot outperform KEPLER-Wiki on NLP tasks (Section 4.2), it shows significant improvements in LAMA-UHN, which suggests that we should explore which kind of knowledge is needed on different scenarios in the future. (3) All the RoBERTa-based models perform worse than vanilla $\text{BERT}_{\text{BASE}}$ by a large margin, which is consistent with the results of Wang et al. (2020). This may be due to different vocabularies used in BERT and RoBERTa, which presents the vulnerability of LAMA-style probing again (Kassner and Schütze, 2020). We will leave developing a better knowledge probing framework as our future work.

5.3 Running Time Comparison

Compared to vanilla PLMs, KEPLER does not introduce any additional parameters or computations during fine-tuning and inference, which is efficient for practice use. We compare the running time of KEPLER and other knowledge-enhanced PLMs (ERNIE and KnowBert) in Table 12. The time is evaluated on TACRED training set for one epoch with one NVIDIA Tesla V100 (32 GB), and all models use 32 batch size and 128 sequence length. The ''entity linking'' time of KnowBert is for entity candidate generation. We can observe that KEPLER requires much less running time since it does not need entity linking or entity embedding fusion, which will benefit time-sensitive applications.

5.4 Correlation with Entity Frequency

To better understand how KEPLER helps the entity-centric tasks, we provide analyses on the correlations between KEPLER performance and entity frequency in this section. The motivation is to verify a natural hypothesis that KEPLER improvements mainly come from better representing the entity mentions in text, especially the rare entities, which do not show up frequently in the pre-training corpora and thus cannot be well learned by the language modeling objectives.

We perform entity linking for the TACRED dataset with BLINK (Wu et al., 2020) to link the entity mentions in text to their corresponding Wikipedia identifiers. Then we count the occurrences of the entities in Wikipedia with the hyperlinks in rich text, denoting the entity frequencies. We conduct two experiments to analyze the correlations between KEPLER performance and entity frequency: (1) In Table 13, we divide the

Model	LAMA				LAMA-UHN	
	Google-RE	T-REx	ConceptNet	SQuAD	Google-RE	T-REx
BERT	9.8	31.1	15.6	14.1	4.7	21.8
RoBERTa	5.3	24.7	19.5	9.1	2.2	17.0
Our RoBERTa	7.0	23.2	**19.0**	8.0	2.8	15.7
KEPLER-Wiki	**7.3**	**24.6**	18.7	**14.3**	3.3	16.5
KEPLER-W+W	**7.3**	24.4	17.6	10.8	**4.1**	**17.1**

Table 11: P@1 results on knowledge probing benchmark LAMA and LAMA-UHN.

Model	Entity Linking	Fine-tuning	Inference
$\text{ERNIE}_{\text{RoBERTa}}$	780s	730s	194s
$\text{KnowBert}_{\text{RoBERTa}}$	190s	677s	235s
KEPLER	**0s**	**508s**	**152s**

Table 12: Three parts of running time for one epoch of TACRED training set.

entity mentions into five parts by their frequencies, and compare the TACRED performances while only keeping entities in one part and masking the other. (2) In Figure 3, we sequentially mask the entity mentions in the ascending order of entity frequencies and see the F-1 changes.

From the results, we can observe that:

(1) Figure 3 shows that when the entity masking rate is low, the improvements of KEPLER over RoBERTa are generally much higher than when the entity masking rate is high. It indicates that the improvements of KEPLER do mainly come from better modeling entities in context. However, even when all the entity mentions are masked, KEPLER still outperforms RoBERTa. We claim this is because the KE objective can also help to learn to understand fact-related text since it requires the model to recall facts from textual descriptions. This claim is further substantiated in Section 5.5.

(2) From Table 13, we can observe that the improvement in the ‘‘0%-20%’’ setting is marginally higher than the other settings, which demonstrates that KEPLER does have special advantages on modeling rare entities compared to vanilla PLMs. But the improvements in the frequent settings are also significant and we cannot say that the overall improvements of KEPLER are mostly from the rare entities. In general, the results in Table 13 show that KEPLER can better model all the entities, no matter rare or frequent.

5.5 Understanding Text or Storing Knowledge

We argue that by jointly training the KE and the MLM objectives, KEPLER (1) can better understand fact-related text and better extract knowledge from text, and also (2) can remember factual knowledge. To investigate the two abilities of KEPLER in a quantitative aspect, we carry out an experiment on TACRED, in which the head and tail entity mentions are masked (masked-entity, ME) or only head and tail entity mentions are shown (only-entity, OE). The ME setting shows to what extent the models can extract facts only from the textual context without the clues in entity names. The OE setting demonstrates to what extent the models can store and predict factual knowledge, as only the entity names are given to the models.

As shown in Table 14, KEPLER-Wiki shows significant improvements over Our RoBERTa in both settings, which suggests that KEPLER has indeed possessed superior abilities on both extracting and storing knowledge compared to vanilla PLMs without knowledge infusion. And the KEPLER-KE model performs poorly on the ME setting but achieves marginal improvements on the OE setting. It indicates that without the help of the MLM objective, KEPLER only learns the entity description embeddings and degenerates in general language understanding, while it can still remember knowledge into entity names to some extent.

6 Related Work

Pre-training in NLP There has been a long history of pre-training in NLP. Early works focus on distributed word representations (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014), many of which are often adopted in current models as word embeddings. These

Entity Frequency	0%-20%	20%-40%	40%-60%	60%-80%	80%-100%
KEPLER-Wiki	64.7	64.4	64.8	64.7	68.8
Our RoBERTa	64.1	64.3	64.5	64.3	68.5
Improvement	+0.6	+0.1	+0.3	+0.4	+0.3

Table 13: F-1 scores on TACRED (%) under different settings by entity frequencies. We sort the entity mentions in TACRED by their corresponding entity frequencies in Wikipedia. The ‘‘0%-20%’’ setting indicates only keeping the least frequent 20% entity mentions and masking all the other entity mentions (for both training and validation), and so on. The results are averaged over 5 runs.

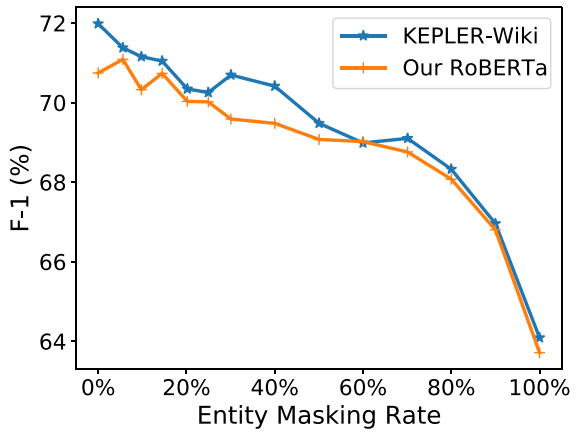

Figure 3: TACRED performance (F-1) of KEPLER and RoBERTa change with the rate of entity mentions being masked.

Model	ME	OE
Our RoBERTa	54.0	46.8
KEPLER-KE	40.2	47.0
KEPLER-Wiki	54.8	48.9

Table 14: Masked-entity (ME) and only-entity (OE) F-1 scores on TACRED (%).

pre-trained embeddings can capture the semantics of words from large-scale corpora and thus benefit NLP applications. Peters et al. (2018) push this trend a step forward by using a bidirectional LSTM to form contextualized word embeddings (ELMo) for richer semantic meanings under different circumstances.

Apart from word embeddings, there is another trend exploring pre-trained language models. Dai and Le (2015) propose to train an auto-encoder on unlabeled textual data and then fine-tune it on downstream tasks. Howard and Ruder (2018) propose a universal language model (ULMFiT). With the powerful Transformer architecture (Vaswani et al., 2017), Radford et al. (2018) demonstrate an effective pre-trained generative model (GPT). Later, Devlin et al. (2019) release a pre-trained deep Bidirectional Encoder Representation from Transformers (BERT), achieving state-of-the-art performance on a wide range of NLP benchmarks.

After BERT, similar PLMs spring up recently. Yang et al. (2019) propose a permutation language model (XLNet). Later, Liu et al. (2019c) show that more data and more parameter tuning can benefit PLMs, and release a new state-of-the-art model (RoBERTa). Other works explore how to add more tasks (Liu et al., 2019b) and more parameters (Raffel et al., 2020; Lan et al., 2020) to PLMs.

Knowledge-Enhanced PLMs Recently, many works have investigated how to incorporate knowledge into PLMs. MTB (Baldini Soares et al., 2019) takes a straightforward ‘‘matching the blank’’ pre-training objective to help the relation classification task. ERNIE (Zhang et al., 2019) identifies entity mentions in text and links pre-processed knowledge embeddings to the corresponding positions, which shows improvements on several NLP benchmarks. With a similar idea as ERNIE, KnowBert (Peters et al., 2019) incorporates an integrated entity linker in their model and adopts end-to-end training. Besides, Logan et al. (2019) and Hayashi et al. (2020) utilize relations between entities inside one sentence to train better generation models. Xiong et al. (2019) adopt entity replacement knowledge learning for improving entity-related tasks.

Some contemporaneous or following works try to inject factual knowledge into PLMs in different ways. E-BERT (Poerner et al., 2020) aligns entity embeddings with word embeddings and then directly adds the aligned embeddings into BERT to avoid additional pre-training. K-Adapter (Wang et al., 2020) injects knowledge with additional neural adapters to support continuous learning.

Knowledge Embedding KE methods have been extensively studied. Conventional KE models define different scoring functions for relational triplets. For example, TransE (Bordes et al., 2013) treats tail entities as translations of head entities and uses L_1-norm or L_2-norm to score triplets, while DistMult (Yang et al., 2015) uses matrix multiplications and ComplEx (Trouillon et al., 2016) adopts complex operations based on it. RotatE (Sun et al., 2019) combines the advantages of both of them.

Inductive Embedding Above KE methods learn entity embeddings only from KG and are inherently transductive, while some works (Wang et al., 2014; Xie et al., 2016; Yamada et al., 2016; Cao et al., 2017; Shi and Weninger, 2018; Cao et al., 2018) incorporate textual metadata such as entity names or descriptions to enhance the KE methods and hence can do inductive KE to some extent. Besides KG, it is also common for general inductive graph embedding methods (Hamilton et al., 2017; Bojchevski and Günnemann, 2018) to utilize additional node features like text attributes, degrees, and so on. KEPLER follows this line of studies and takes full advantage of textual information with an effective PLM.

Hamaguchi et al. (2017) and Wang et al. (2019c) perform inductive KE by aggregating the trained embeddings of the known neighboring nodes with graph neural networks, and thus do not need additional features. But these methods require the unseen nodes to be surrounded by known nodes and cannot embed new (sub)graphs. We leave how to develop KEPLER to do fully inductive KE without additional features as future work.

7 Conclusion and Future Work

In this paper, we propose KEPLER, a simple but effective unified model for knowledge embedding and pre-trained language representation. We train KEPLER with both the KE and MLM objectives to align the factual knowledge and language representation into the same semantic space, and experimental results on extensive tasks demonstrate its effectiveness on both NLP and KE applications. Besides, we propose Wikidata5M, a large-scale KG dataset to facilitate future research.

In the future, we will (1) explore advanced ways for more smoothly unifying the two semantic space, including different KE forms and different training objectives, and (2) investigate better knowledge probing methods for PLMs to shed light on knowledge-integrating mechanisms.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503), the National Natural Science Foundation of China (NSFC No. U1736204, 61533018, 61772302, 61732008), grants from Institute for Guo Qiang, Tsinghua University (2019GQB0003), and Beijing Academy of Artificial Intelligence (BAAI2019ZD0502). Prof. Jian Tang is supported by the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant and the Canada CIFAR AI Chair Program. Xiaozhi Wang and Tianyu Gao are supported by Tsinghua University Initiative Scientific Research Program. We also thank our action editor, Prof. Doug Downey, and the anonymous reviewers for their consistent help and insightful suggestions.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of EMNLP-IJCNLP*, pages 5185–5194. **DOI:** https://doi.org/10.18653/v1/D19-1522

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL*, pages 2895–2905. **DOI:** https://doi.org/10.18653/v1/P19-1279

Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *Proceedings of ICLR*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2787–2795.

Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Chengjiang Li, Xu Chen, and Tiansi Dong.

2018. Joint representation learning of cross-lingual words and entities via attentive distant supervision. In *Proceedings of EMNLP*, pages 227–237. **DOI:** https://doi.org/10.18653/v1/D18-1021

Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. 2017. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *Proceedings of ACL*, pages 1623–1633. **DOI:** https://doi.org/10.18653/v1/P17-1149

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of ACL*, pages 87–96.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3079–3087.

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.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of EMNLP-IJCNLP*, pages 6251–6256.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. In *Proceedings of IJCAI*, pages 1802–1808. **DOI:** https://doi.org/10.24963/ijcai.2017/250

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1025–1035.

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. **DOI:** https://doi.org/10.18653/v1/D18-1514

Hiroaki Hayashi, Zecong Hu, Chenyan Xiong, and Graham Neubig. 2020. Latent relation language models. In *Proceedings of AAAI*, pages 7911–7918. **DOI:** https://doi.org/10.1609/aaai.v34i05.6298

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of ACL*, pages 328–339. **DOI:** https://doi.org/10.18653/v1/P18-1031, **PMID:** 28889062

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of ACL*, pages 7811–7818. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.698

Seyed Mehran Kazemi and David Poole. 2018. SimplE embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4284–4295.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of ICLR*.

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.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferabilityof

contextual representations. In *Proceedings of NAACL-HLT*, pages 1073–1094.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of AAAI*, pages 2901–2908. **DOI:** https://doi.org/10.1609/aaai.v34i03.5681

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. In *Proceedings of ACL*, pages 4487–4496.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, cs.CL/1907.11692v1.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's Wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of ACL*, pages 5962–5971. **DOI:** https://doi.org/10.18653/v1/P19-1598

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of ACL*, pages 3449–3460. **DOI:** https://doi.org/10.18653/v1/P19-1335

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and motivation*, volume 24, pages 109–165. Elsevier. **DOI:** https://doi.org/10.1016/S0079-7421(08)60536-8

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41. **DOI:** https://doi.org/10.1145/219717.219748

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT (Demonstrations)*, pages 48–53. **DOI:** https://doi.org/10.18653/v1/N19-4009

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543. **DOI:** https://doi.org/10.3115/v1/D14-1162

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237. **DOI:** https://doi.org/10.18653/v1/N18-1202

Matthew E. Peters, Mark Neumann, Robert 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. **DOI:** https://doi.org/10.18653/v1/D19-1005, **PMID:** 31383442

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-IJCNLP*, pages 2463–2473. **DOI:** https://doi.org/10.18653/v1/D19-1250

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818. **DOI:** https://doi.org/10.18653/v1/2020.findings-emnlp.71

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving

language understanding by generative pre-training. In *Technical report, OpenAI*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725. **DOI:** https://doi.org/10.18653/v1/P16-1162

Baoxu Shi and Tim Weninger. 2018. Open-world knowledge graph completion. In *Proceedings of AAAI*, pages 1957–1964.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4077–4087.

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

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you Need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? Sentence-level pretraining beyond language modeling. In *Proceedings of ACL*, pages 4465–4476. **DOI:** https://doi.org/10.18653/v1/P19-1439

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of ICLR*. **DOI:** https://doi.org/10.18653/v1/W18-5446

PeiFeng Wang, Jialong Han, Chenliang Li, and Rong Pan. 2019c. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *Proceedings of AAAI*, pages 7152–7159. **DOI:** https://doi.org/10.1609/aaai.v33i01.33017152

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Cuihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-Adapter: Infusing knowledge into pre-trained models with adapters. *CoRR*, cs.CL/2002.01808v3.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*, pages 1591–1601. **DOI:** https://doi.org/10.3115/v1/D14-1167

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122. **DOI:** https://doi.org/10.18653/v1/N18-1101

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of EMNLP*, pages 6397–6407.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of AAAI*, pages 2659–2665.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Stoyanov Veselin. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *Proceedings of ICLR*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of CoNLL*, pages 250–259. **DOI:** https://doi.org/10.18653/v1/K16-1025

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of ACL*, pages 1436–1446. **DOI:** https://doi.org/10.18653/v1/P17-1132

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5754–5764.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of ACL*, pages 656–661. **DOI:** https://doi.org/10.18653/v1/P18-2104

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*, pages 35–45. **DOI:** https://doi.org/10.18653/v1/D17-1004

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. **DOI:** https://doi.org/10.18653/v1/P19-1139

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of ICCV*, pages 19–27. **DOI:** https://doi.org/10.1109/ICCV.2015.11

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.