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

**Xiaozhi Wang**[1], **Tianyu Gao**[1], **Zhaocheng Zhu**[2,3], **Zhiyuan Liu**[1], **Juanzi Li**[1], **Jian Tang**[2,4,5]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Mila - Québec AI Institute, Montréal, Canada
[3]Univesité de Montréal, Montréal, Canada
[4]HEC, Montréal, Canada
[5]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 do not utilize the rich text data. 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 effectively learn KE through the abundant information in text. In KEPLER, we encode textual descriptions of entities 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 performance on various NLP tasks, and also works remarkably well as an inductive KE model on the link prediction task. Furthermore, for pre-training KEPLER and evaluating the KE performance, we construct Wikidata5M, 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 dataset can be obtained from 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., 2019d), learn effective language representation from large-scale unstructured corpora with unsupervised language modeling objectives. They have achieved superior performance on various natural language processing (NLP) tasks.

Preprint. Work in progress.

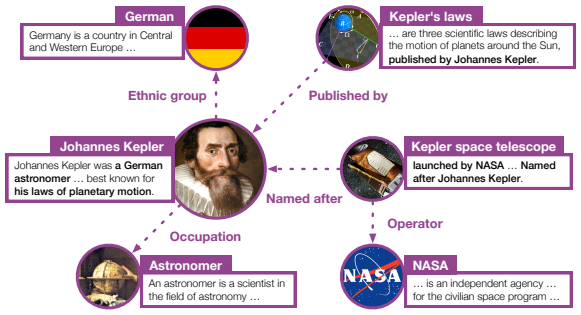


Figure 1: An example of a KG with entity descriptions. The figure suggests that descriptions contain lots of information about entities and can help to predict the relational facts between them.

Existing PLMs learn useful linguistic knowledge from unlabeled text (Liu et al., 2019a), but they generally cannot well capture the world facts, which are crucial for many NLP tasks, but are typically sparse and have complex and diverse 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 efficiently embed them into continuous vectors of entities and relations. These embeddings not only help with the KG completion task, but can also improve the performance of various NLP applications (Zaremoodi et al., 2018; Zhong et al., 2019). However, conventional KE models solely take the KG structures as input and do not involve textual data, and thus cannot directly help the pre-training of PLMs.

Inspired by Xie et al. (2016), we take entity descriptions to bridge the gap between KE and PLM. As shown in Figure 1, textual descriptions contain abundant information about entities, which helps to align the semantic space of text to the symbol space of KGs. In this way, KE methods can provide factual knowledge for PLMs, while the informative text data can also benefit KE.

In this paper, we propose **KEPLER**, a unified model for **K**nowledge **E**mbedding and **P**re-trained **L**anguag**E** **R**epresentation. We encode the texts 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 during pre-training. For the KE objective, we encode the entity descriptions as their corresponding entity embeddings, and then learn 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., 2019d). 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 descriptive texts. (4) KEPLER can be directly adopted in a wide range of NLP tasks without additional inference overhead compared to conventional PLMs, since we do not modify model structures but add new training objectives.

There are also some recent works (Zhang et al., 2019; Peters et al., 2019; Liu et al., 2019b) that directly add fixed entity embeddings into PLMs for providing external factual 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 words in context to the corresponding entities, making them suffer from the error propagation problem. (3) Compared with vanilla PLMs, their sophisticated mechanisms to retrieve and use entity representations lead to additional inference overhead.

**As a KE model**, (1) KEPLER can better utilize the abundant information from entity descriptions due to the help of the MLM objective. (2) KEPLER is capable of performing KE in the inductive setting, i.e., it can get the embeddings for unseen entities from their descriptions, while conventional KE methods are inherently transductive and they can only learn representations for the entities appearing in the training datasets. Inductive KE is essential for many real-world applications, such as updating KGs with new 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 triples, and aligned entity descriptions from Wikipedia. To the best of our knowledge, it is the first million-scale general-domain KG dataset. We also benchmark several classical KE methods and give data splits for both the transductive and the inductive setting to facilitate future research.

To summarize, our contribution is three-fold: (1) We propose KEPLER, a knowledge-enhanced PLM with 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 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 pre-trained embeddings can capture 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) based on AWD-LSTM (Merity et al., 2018). 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.

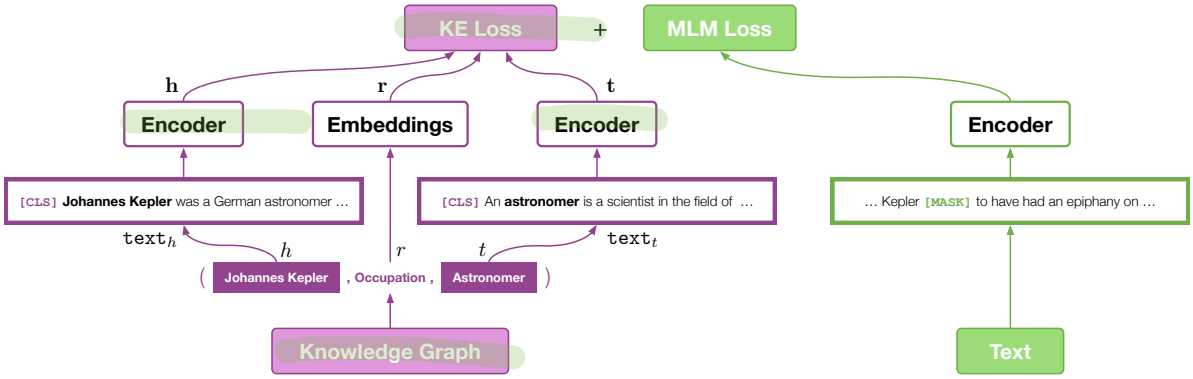


Figure 2: A demonstration for the structure of KEPLER. By jointly training with the knowledge embedding (KE) and the masked language modeling (MLM) objectives, our framework can implicitly incorporate knowledge into the language representation model.

After BERT, similar PLMs spring up recently. Yang et al. (2019) propose a permutation language model (XLNet). Later, Liu et al. (2019d) 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., 2019c) and more parameters (Raffel et al., 2019; 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); Hayashi et al. (2020) utilize relations between entities inside one sentence to help train better generation models, and Xiong et al. (2019) adopt entity replacement knowledge learning for improving entity-related tasks.

**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 from both of them.

Above models typically learn entity embeddings from KG structures, while some works (Wang et al., 2014; Xie et al., 2016; Yamada et al., 2016; Cao et al., 2017, 2018) incorporate textual metadata such as entity names or entity descriptions to enhance the KE methods. Though the motivations are similar to ours, these works focus on improving KG completion with external textual information, while KEPLER can also benefit a wide range of NLP applications as a knowledge-enhanced PLM.

## 3 KEPLER

As shown in Figure 2, KEPLER is a unified model for knowledge embedding and pre-trained language representation. It incorporates both factual knowledge and language understanding into one PLM by jointly training with two objectives. In this section, we introduce the encoder structure (Section 3.1), how we train KEPLER with the knowledge embedding (Section 3.2) and the masked language modeling (Section 3.3) objectives, and how we combine the two as a unified model (Section 3.4).

### 3.1 Encoder

For the text encoder, we use Transformer architecture (Vaswani et al., 2017) in the same way as Devlin et al. (2019); Liu et al. (2019d). 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 the combination of a multi-head self-attention and a multi-layer 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 `[CLS]` added to the beginning of the text (Devlin et al., 2019), and the output at `[CLS]` is regarded as the representation for the whole sentence. Denote the representation as $\texttt{E}_{\texttt{[CLS]}}(\texttt{text})$.

In the encoder, tokenization is to convert plain texts into sequences of tokens. Here we use the same tokenizer as in RoBERTa: the Byte-Pair Encoding (BPE) (Sennrich et al., 2016). It performs better than the subword tokenization used in BERT.

Unlike previous knowledge-enhanced models based on PLMs (Zhang et al., 2019; Peters et al., 2019), we do not modify the Transformer encoder structure. That is to say, we do not add external entity linkers or knowledge-integration layers to KEPLER. 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 BERT or RoBERTa.

### 3.2 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 formally 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 an edge, 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 choices for the KE part of KEPLER. Here we introduce two simple but effective ways: only using entity descriptions, and using both entity and relation descriptions.

**Using Entity Descriptions** For a relational triplet $(h, r, t)$, we have:

$$\begin{aligned} \mathbf{h} &= \texttt{E}_{\texttt{[CLS]}}(\texttt{text}_h), \\ \mathbf{t} &= \texttt{E}_{\texttt{[CLS]}}(\texttt{text}_t), \\ \mathbf{r} &= \mathbf{T}_r, \end{aligned} \tag{1}$$

where $\texttt{text}_h$ and $\texttt{text}_t$ are the descriptions for $h$ and $t$, concatenated with a special `[CLS]` token at the beginning. $\texttt{E}_{\texttt{[CLS]}}$ is the output of the text encoder at the position of `[CLS]`. $\mathbf{T} \in \mathbb{R}^{|\mathcal{R}| \times d}$ is the relation embeddings and $\mathbf{h}, \mathbf{t}, \mathbf{r}$ are the embeddings for entities $h, t$ and the relation $r$.

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

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

where $(h'_i, r, t'_i)$ are negative samples, $\gamma$ is the margin, $\sigma$ 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,

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

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.

**Using Entity and Relation Descriptions** In this one, we use entity embeddings conditioned on $r$. 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,

$$\mathbf{h}_r = \texttt{E}_{\texttt{[CLS]}}(\texttt{text}_{h,r}), \tag{4}$$

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 `[CLS]` at the beginning and `[SEP]` in between. Correspondingly, we use $\mathbf{h}_r$ instead of $\mathbf{h}$ for Equation 2, 3.

### 3.3 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 another random token, 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.

| Dataset | #entity | #relation | #training | #validation | #test |
|---|---|---|---|---|---|
| FB15K | 14,951 | 1,345 | 483,142 | 50,000 | 59,071 |
| WN18 | 40,943 | 18 | 141,442 | 5,000 | 5,000 |
| FB15K-237 | 14,541 | 237 | 272,115 | 17,535 | 20,466 |
| WN18RR | 40,943 | 11 | 86,835 | 3,034 | 3,134 |
| Wikidata5M | 4,818,298 | 822 | 21,343,681 | 5,357 | 5,321 |

Table 1: Statistics of Wikidata5M compared with existing widely-used KE benchmarks.

| Entity Type | Occurrence | Percentage |
|---|---|---|
| Human | 1,517,591 | 31.5% |
| Taxon | 363,882 | 7.6% |
| Film | 114,266 | 2.4% |
| Human Settlement | 110,939 | 2.3% |
| Total | 2,106,678 | 43.8% |

Table 2: Top-4 entity categories in Wikidata5M.

| 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.

We initialize our model with the pre-trained checkpoint of RoBERTa$_{\text{BASE}}$. 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 our experiments, only using the KE objective leads to poor results in NLP tasks.

### 3.4 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 5,

$$\mathcal{L} = \mathcal{L}_{\text{KE}} + \mathcal{L}_{\text{MLM}}, \tag{5}$$

where $\mathcal{L}_{\text{KE}}$ and $\mathcal{L}_{\text{MLM}}$ 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.

## 4 Wikidata5M

As shown in Section 3, to train KEPLER, we need (1) a large-scale KG, and (2) the corresponding descriptions for its entities and relations. Also, we need a KG dataset (3) with an inductive setting for investigating this crucial direction of KG applications, which most existing KG datasets do not have. Thus, based on Wikidata and Wikipedia, we construct Wikidata5M, a new 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 steps (Section 4.1) and the data split (Section 4.2), and then provide the results of popular KE methods on this dataset (Section 4.3).

### 4.1 Data Collection

We pull the dumps of Wikidata[1] and Wikipedia[2] from their websites respectively. We remove pages whose first paragraphs contain fewer than five words in Wikipedia. For each entity in Wikidata, we align it to its Wikipedia page. The first sections of Wikipedia pages are extracted as the descriptions for entities. Entities that have no corresponding Wikipedia pages are discarded.

To construct the KG, we retrieve all the relational statements in Wikidata, where entities and relations are linked to their canonical IDs. A statement is considered to be valid if both of its entities can be aligned with Wikipedia pages, and its relation has a non-empty page in Wikidata. The final KG contains 4,818,298 entities, 822 relations and 21,343,681 triplets. Statistics of the Wikidata5M dataset compared with four other widely-used datasets are shown in Table 1. Top-4 entity categories are listed in Table 2. We can see that our Wikidata5M is much larger than existing KG datasets, covering all sorts of domains.

[1] https://www.wikidata.org
[2] https://en.wikipedia.org

| 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: Performances of different KE models on Wikidata5M (%).

### 4.2 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 should learn good entity embeddings during training. In the **inductive setting** (shown in Table 3), the entities and triplets are mutually disjoint across training and test, which means during inference, KE models need to form the entity embeddings inductively. 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.

### 4.3 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 the multi-GPU toolkit GraphVite (Zhu et al., 2019).

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 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 benchmark of popular KE methods on Wikidata5M. Compared to classical datasets, Wikidata5M is more challenging due to its large scale and high coverage on different types of entities and relations. The results advocate for more efforts towards large-scale KGs.

## 5 Experiments

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

### 5.1 Pre-training Settings

In experiments, we choose RoBERTa (Liu et al., 2019d) 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 and use the released `roberta.base`[3] parameters for initialization.

**KE Objective** For the KE objective, we use the following three different settings:

(1) **KEPLER-Wiki** In this setting, we use Wikidata5M as our pre-training KG source. As stated in Section 4.1, it contains descriptions for all the entities, and we can acquire the descriptive texts for relations from the Wikidata website. We always take the first 512 tokens from the description as the input. Additionally, we denote the case of using both entity and relation descriptions (as described in Equation 4) as **KEPLER-Wiki-rel**, which is almost the same as KEPLER-Wiki except the head entity description is concatenated with the corresponding relation description.

(2) **KEPLER-WordNet** WordNet (Miller, 1995) is an English lexical graph, where nodes are lemmas and synsets, and edges are their relations. Intuitively, incorporating WordNet can bring our model more lexical knowledge and thus benefits NLP tasks. We use the WordNet 3.0 the same as in Peters et al. (2019), which is extracted from the `nltk`[4] package. Descriptions for both lemmas and synsets are provided in the metadata. Since the number of relations is relatively small , we do not adopt the relation description setting here.

[3] https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.md

[4] https://www.nltk.org/

(3) **KEPLER-W+W** In this setting, we take both Wikidata5M and WordNet as the KG sources. To jointly train with the two KG datasets, we modify the objective in Equation 5 as

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

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

For all KE settings, we take the following negative sampling policy: For each triplet $(h, r, t)$, we sample a negative head entity $h'$ and a negative tail entity $t'$, and then use Equation 2 for training.

**MLM Objective** For the MLM objective, we use the BookCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words) as our pre-training corpora. We extract text from these two sources in the same way as Devlin et al. (2019).

**Hyperparameters** Since KEPLER is based on the implementation of RoBERTa, we adopt the hyperparameter settings for PLM from `fairseq`. As for KE, we set different margins $\gamma$ (in Equation 2) for different KGs and different downstream tasks. For Wikidata5M, we set $\gamma = 2$ for NLP downstream tasks and $\gamma = 9$ for KG tasks. For WordNet, we set $\gamma = 1$ for NLP tasks.

**Baselines** Since KEPLER is based on **RoBERTa** and we use the same training framework as it does, $\text{RoBERTa}_{\texttt{BASE}}$ is one of our primary baselines. However, we do not have the full training corpora of RoBERTa (126GB, and we only have 13GB), so for fair comparisons, we also evaluate **RoBERTa**$^*$, which is initialized by the $\text{RoBERTa}_{\texttt{BASE}}$ checkpoint, and is further trained on the same corpora as KEPLER with only the MLM objective.

Besides, we evaluate several recent approaches on knowledge-enhanced PLMs, including ERNIE (Zhang et al., 2019), KnowBERT (Peters et al., 2019) and MTB (Baldini Soares et al., 2019). Note that ERNIE and KnowBERT are based on $\text{BERT}_{\texttt{BASE}}$ and MTB is based on $\text{BERT}_{\texttt{LARGE}}$. Also, MTB uses "matching the blank" pre-training, which specifically targets the relation classification scenario, while KEPLER, ERNIE and KnowBERT adopt general knowledge-enhanced techniques.

### 5.2 NLP Tasks

In this section, we introduce how KEPLER can be used as a knowledge-enhanced PLM on various NLP tasks and show its performance compared with state-of-the-art models.

| Model | P | R | F-1 |
|---|---|---|---|
| BERT | 67.2 | 64.8 | 66.0 |
| $\text{BERT}_{\texttt{LARGE}}$ | - | - | 70.1 |
| RoBERTa | 71.1 | 70.5 | 70.8 |
| ERNIE | 70.0 | 66.1 | 68.0 |
| MTB | - | - | 71.5 |
| KnowBERT | 71.6 | 71.4 | 71.5 |
| RoBERTa$^*$ | 71.3 | 69.8 | 70.5 |
| KEPLER-Wiki | 72.8 | 72.2 | **72.5** |
| KEPLER-WordNet | 73.0 | 69.3 | 71.1 |
| KEPLER-W+W | 72.5 | 72.1 | 72.3 |

Table 5: Precision, recall and F-1 results on TACRED (%). Except $\text{BERT}_{\texttt{LARGE}}$ and MTB, all other models use the `BASE` size. BERT and $\text{BERT}_{\texttt{LARGE}}$ results are reported in Zhang et al. (2019); Baldini Soares et al. (2019). Other baseline results are from their corresponding papers.

#### Relation Classification

Relation classification is an important NLP task that requires models to classify relation types between two given entities from text. We evaluate KEPLER and other baseline models on two commonly-used relation classification datasets:

(1) **TACRED** (Zhang et al., 2017) is a human-annotated relation classification dataset covering 42 relation types and 106,264 sentences. Here we follow the fine-tuning procedure of Zhang et al. (2019), where four special tokens are added before and after the two entity mentions in the sentence to highlight the entity positions.

Table 5 shows the evaluation results of various models on TACRED, from which we can see that KEPLER-Wiki achieves the new state-of-the-art on the benchmark. KEPLER has a great performance promotion over RoBERTa$^*$ (our baseline model) and also shows improvements compared to other methods, even if they use a `LARGE` architecture ($\text{BERT}_{\texttt{LARGE}}$ and MTB). Besides, KEPLER-WordNet also shows a slight improvement over RoBERTa$^*$, while KEPLER-W+W achieves comparable results with KEPLER-Wiki. It suggests that pre-training with WordNet has limited benefits and combining the two KGs cannot bring better performance in our framework.

(2) **FewRel** (Han et al., 2018) is a few-shot relation classification dataset with 100 relations and 70,000 instances. Based on it, Gao et al. (2019) propose FewRel 2.0, which adds the new domain adaptation (DA) challenge with a new test set.

| 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 (RoBERTa) | 85.78 | 95.78 | 77.65 | 92.26 | 64.65 | 82.76 | 50.80 | 71.84 |
| Proto (RoBERTa*) | 84.42 | 95.30 | 76.43 | 91.74 | 61.98 | 83.11 | 48.56 | 72.19 |
| 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 (RoBERTa) | 89.32 | 93.70 | 82.49 | 88.43 | 66.78 | 81.84 | 53.99 | 70.85 |
| PAIR (RoBERTa*) | 89.26 | 93.71 | 83.32 | 89.02 | 63.22 | 77.66 | 49.28 | 65.97 |
| 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. "Proto" indicates Prototypical Networks (Snell et al., 2017), "PAIR" is from Gao et al. (2019) and "MTB" is from Baldini Soares et al. (2019). MTB uses the LARGE size and all the other models use the BASE size.

| Model | P | R | F-1 |
|---|---|---|---|
| UFET | 77.4 | 60.6 | 68.0 |
| BERT | 76.4 | 71.0 | 73.6 |
| RoBERTa | 77.4 | 73.6 | 75.4 |
| ERNIE | 78.4 | 72.9 | 75.6 |
| KnowBERT | 78.6 | 73.7 | 76.1 |
| RoBERTa* | 75.1 | 73.4 | 74.3 |
| KEPLER-Wiki | 77.8 | 74.6 | **76.2** |

Table 7: Entity typing results on OpenEntity (%). All methods with PLMs use the BASE size.

Few-shot relation classification takes the $N$-way $K$-shot setting. Relations in the training set and the test set of FewRel are disjoint, and 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 the query sentences into one of the $N$ relations based on the sampled $N \times K$ instances.

In our evaluation, we use two state-of-the-art frameworks: Prototypical Networks (Snell et al., 2017) and PAIR (Gao et al., 2019). We replace the text encoders in the two frameworks with our baseline models and KEPLER, and compare their performance. Since FewRel is constructed on Wikidata, to avoid information leak of the test set, we delete all the triplets shown in the FewRel test set from our Wikidata5M training source.

As shown in Table 6, for both frameworks, our models have superior performance over the BASE-size PLMs. We also compare our model with MTB (Baldini Soares et al., 2019). Note that MTB (1) uses the LARGE architecture of BERT and (2) has a test set leaking problem since it performs pre-training over the whole Wikipedia and Wikidata, which are the source of FewRel.

From the results, we also have two interesting observations: (1) RoBERTa shows superior results over BERT on almost every benchmark, yet PAIR (BERT) is comparable with PAIR (RoBERTa), and even better under some settings. Since the PAIR model uses sentence concatenation, this may indicate that the next sentence prediction (NSP) objective, which is used in BERT but discarded in RoBERTa, is useful for downstream tasks requiring sentence concatenation. (2) KEPLER not only shows improvements on FewRel 1.0, but also brings promotion on FewRel 2.0, which involves a medical KG that is not included in our training sources. It suggests that the pre-training of KEPLER not only learns a good representation for entities (which is crucial for relation classification tasks), but also gains a better understanding of text and acquires a general ability to extract factual knowledge from the context.

### 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).

To identify the entity mentions of interest, we add two special tokens before and after the entity spans, and use the representation of the first special token as the feature in the final classification step. As shown in Table 7, compared to our baseline RoBERTa*, KEPLER achieves an improvement of 1.9% and leads to the state-of-the-art result. Note

| Model | MNLI (m/mm) 392K | QQP 363K | QNLI 104K | SST-2 67K |
|---|---|---|---|---|
| RoBERTa | 87.5/87.3 | 91.9 | 92.8 | 94.8 |
| RoBERTa* | 87.1/86.9 | 90.9 | 92.4 | 94.7 |
| KEPLER | 87.2/86.5 | 91.5 | 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 | 78.7 |
| RoBERTa* | 63.4 | 89.9 | 88.2 | 78.0 |
| KEPLER | 62.3 | 89.4 | 89.3 | 70.8 |

Table 8: GLUE results on the `dev` set (%).

that KEPLER does not perform any kind of linking, yet ERNIE and KnowBERT use entity linking and entity embeddings, which gives them a specific edge in entity-centered tasks. Still, KEPLER performs the best among its counterparts.

### GLUE

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) is a collection of several natural language understanding tasks and is often used in evaluating PLMs. These tasks generally do not require knowledge-enhanced language understanding (Zhang et al., 2019), so to see whether the performance of our model decreases on these tasks, we evaluate KEPLER on GLUE and compare it with other baselines.

Table 8 shows the results on GLUE. We notice that KEPLER keeps a consistent result with its baseline on large datasets like MNLI, but suffers an unstable performance on small datasets like RTE. In general, KEPLER achieves comparable results with RoBERTa* on GLUE, which suggests that the joint training in KEPLER does not harm the general language understanding ability.

## 5.3 KG Tasks

In this section, we show how KEPLER works as a KE model, and evaluate it on our Wikidata5M dataset in both the transductive link prediction setting and the inductive setting.

We do not use the existing KE benchmarks because (1) they are lack of high-quality text descriptions for their entities and (2) they do not have a reasonable data split for the inductive setting.

### Transductive Setting

For the transductive setting, all the entities have been seen during the training phase, so the model can learn good representations for these entities. The test triplets and the training triplets, however, are disjoint. Good performance in the transductive link prediction task shows good properties of learned entity embeddings, since they reflect the relationship between entities. For KEPLER, we acquire the entity embeddings by encoding their descriptions, following Equation 1 and 4.

Table 9a shows the results of KEPLER and our baseline model TransE in the transductive setting. Though TransE outperforms KEPLER, it is a reasonable result: (1) KEPLER is a pre-training model targeting both KE and MLM, with no specific fine-tuning for the link prediction task, and (2) while training entity embeddings is a more direct method towards the link prediction task, KEPLER does not store entity embeddings and it calculates the entity representations from their corresponding descriptions. Also, (3) due to the large model size of KEPLER, it is hard to use a large negative sampling size (we use 1 in KEPLER, but typical KE methods use 64 or more), which is crucial for KE performance (Zhu et al., 2019). Still, KEPLER achieves a favorable performance on this benchmark, demonstrating its strong capacity in KE.

### Inductive Setting

In the inductive setting, both entities and triplets in the test set have not been seen during the training phase. Conventional KE methods cannot work in the inductive setting, because they cannot provide the embeddings for the unseen entities. Thus, we take DKRL (Xie et al., 2016) as our baseline, which also utilizes entity descriptions to calculate entity representations in the inductive setting.

Table 9b shows the inductive results on Wikidata5M. In this setting, KEPLER shows a significant improvement over DKRL, demonstrating the effectiveness of our joint training objective. Besides, KEPLER-Wiki-rel outperforms KEPLER-Wiki by a large margin, indicating that incorporating relation descriptions when encoding the entities brings better performance in link prediction.

# 6 Analysis

### Ablation Study

As shown in Equation 5, KEPLER takes a multi-task loss as its training objective. To demonstrate the effectiveness of the joint objective, we carry out experiments on training with only the MLM loss (RoBERTa*) and only the KE loss (KEPLER-KE), and evaluate them on TACRED. As demon-

| Model | MR | MRR | HITS@1 | HITS@3 | HITS@10 |
|---|---|---|---|---|---|
| TransE (Bordes et al., 2013) | 109370 | 25.3 | 17.0 | 31.1 | 39.2 |
| KEPLER-Wiki | 14454 | 15.4 | 10.5 | 17.4 | 24.4 |
| KEPLER-Wiki-rel | 20267 | 21.0 | 17.3 | 22.4 | 27.7 |

(a) Transductive results on Wikidata5M (%).

| Model | MR | MRR | HITS@1 | HITS@3 | HITS@10 |
|---|---|---|---|---|---|
| DKRL (Xie et al., 2016) | 78 | 23.1 | 5.9 | 32.0 | 54.6 |
| KEPLER-Wiki | 32 | 35.1 | 15.4 | 46.9 | 71.9 |
| KEPLER-Wiki-rel | 28 | 40.2 | 22.2 | 51.4 | 73.0 |

(b) Inductive results on Wikidata5M (%).

Table 9: Link prediction results on Wikidata5M. In the transductive setting, KEPLER gets a reasonable performance. In the inductive setting, KEPLER outperforms the baseline model by a large margin. Across both settings, KEPLER with relation descriptions achieves better results.

| Model | P | R | F-1 |
|---|---|---|---|
| RoBERTa | 71.1 | 70.5 | 70.8 |
| RoBERTa* | 71.3 | 69.8 | 70.5 |
| KEPLER-KE | 66.1 | 63.5 | 64.8 |
| KEPLER-Wiki | 72.8 | 72.2 | 72.5 |

Table 10: Ablation study on TACRED (%). RoBERTa* and KEPLER-KE indicate only using the MLM and the KE objective, respectively.

| Model | ME | OE |
|---|---|---|
| RoBERTa* | 47.0 | 53.7 |
| KEPLER-Wiki | 49.0 | 55.1 |

Table 11: Masked-entity (ME) and only-entity (OE) F-1 results on TACRED (%). We regard that ME result represents text understanding abilities and OE is closer to the KE setting.

strated in Table 10, compared to RoBERTa, both RoBERTa* and KEPLER-KE suffer a performance drop (note that RoBERTa* uses the same training protocol as RoBERTa and the difference is that it uses less training corpora). It suggests that the performance gain of KEPLER is credit to the joint training towards both objectives, which enhances KEPLER with factual knowledge while keeping the strong language understanding ability.

**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 language, and also (2) can remember knowledge and work like a KE model. To investigate the two abilities of KEPLER in a quantitative aspect, we carry out an experiment based on TACRED, in which the entity mentions are masked (masked-entity) or there are only entity mentions (only-entity). We regard that the masked-entity setting shows how well models can extract facts only from the textual context and the only-entity setting demonstrates how well models can store and predict factual knowledge like in the KE task.

As shown in Table 11, KEPLER-Wiki shows a significant improvement over RoBERTa* on both the masked-entity and only-entity settings, which suggests that KEPLER has indeed possessed superior abilities on both extracting and storing knowledge compared to vanilla PLMs.

## 7 Conclusion

In this paper, we propose KEPLER, a unified model for knowledge embedding and pre-trained language representation. We train KEPLER with both the KE and MLM objectives, and experimental results on extensive tasks demonstrate the effectiveness of our model on both NLP and KG applications. Besides, we propose Wikidata5M, a large-scale KG dataset to advance future research.

In the future, we will (1) explore more possible ways of integrating factual knowledge into PLMs, including different forms of knowledge embeddings and different training objectives, for more smoothly unifying the two semantic space. Furthermore, we will also (2) investigate ways of probing the storage of knowledge in models to better guide the research in knowledge-enhanced PLMs.

## References


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.

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.

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.

Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. 2017. Bridge text and knowledge by learcaoning multi-prototype entity mention embedding. In *Proceedings of ACL*, pages 1623–1633.

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 *Proceedings of 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.

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

Hiroaki Hayashi, Zecong Hu, Chenyan Xiong, and Graham Neubig. 2020. Latent relation language models. In *Proceedings of AAAI*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of ACL*, pages 328–339.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Proceedings of 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 transferability of 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. 2019b. K-bert: Enabling language representation with knowledge graph. In *Proceedings of AAAI*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019c. 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. 2019d. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

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.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing lstm language models. In *Proceedings of ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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.

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

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.

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.

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.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. In *Proceedings of Technical report, OpenAI*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of NeurIPS*, 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 *Proceedings of NeurIPS*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of ICLR*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*, pages 1591–1601.

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.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of 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.

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.

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.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019. Improving question answering by commonsense-based pretraining. In *Proceedings of NLPCC*, pages 16–28.

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.

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.