TRELM: Towards Robust and Efficient Pre-training for Knowledge-Enhanced Language Models

Junbing Yan[1,2], **Chengyu Wang**[2], **Taolin Zhang**[2], **Xiaofeng He**[1], **Jun Huang**[2], **Longtao Huang**[2], **Hui Xue**[2], **Wei Zhang**[1]

[1] School of Computer Science and Technology, East China Normal University

[2] Alibaba Group

{junbingyan531,zhangwei.thu2011}@gmail.com, hexf@cs.ecnu.edu.cn

{chengyu.wcy,zhangtaolin.ztl,huangjun.hj,kaiyang.hlt,hui.xueh}@alibaba-inc.com

Abstract

KEPLMs are pre-trained models that utilize external knowledge to enhance language understanding. Previous language models facilitated knowledge acquisition by incorporating knowledge-related pre-training tasks learned from relation triples in knowledge graphs. However, these models do not prioritize learning embeddings for entity-related tokens. Moreover, updating the entire set of parameters in KEPLMs is computationally demanding. This paper introduces **TRELM**, a Robust and Efficient Pre-training framework for Knowledge-Enhanced Language Models. We observe that entities in text corpora usually follow the long-tail distribution, where the representations of some entities are suboptimally optimized and hinder the pre-training process for KEPLMs. To tackle this, we employ a robust approach to inject knowledge triples and employ a knowledge-augmented memory bank to capture valuable information. Furthermore, updating a small subset of neurons in the feed-forward networks (FFNs) that store factual knowledge is both sufficient and efficient. Specifically, we utilize dynamic knowledge routing to identify knowledge paths in FFNs and selectively update parameters during pre-training. Experimental results show that TRELM reduces pre-training time by at least 50% and outperforms other KEPLMs in knowledge probing tasks and multiple knowledge-aware language understanding tasks.

Keywords: Knowledge-Enhanced PLM, Training Efficiency, Robust Pre-trained Language Model

1. Introduction

Pre-trained language models (PLMs) such as BERT (Kenton and Toutanova, 2019) and RoBERTa (Liu et al., 2019) learn language representations from large-scale text corpora and significantly improve the performance of various NLP tasks (Xu et al., 2021; Chang et al., 2021). Yet, they often lack methods for incorporating external knowledge for language understanding (Colon-Hernandez et al., 2021; Cui et al., 2021). Since knowledge graphs (KGs) can provide rich structured knowledge facts (Yang and Mitchell, 2017; Zaremoodi et al., 2018; Han et al., 2018), the performance of PLMs can be enhanced by injecting external knowledge triples from KGs, known as Knowledge-Enhanced PLMs (KEPLMs). KEPLMs (Zhang et al., 2019; Wang et al., 2021b; Sun et al., 2020; Zhang et al., 2022b) incorporate knowledge-related tasks, such as denoising entity auto-encoder (dEA) and knowledge embedding learning, to facilitate knowledge understanding in the models. Figure 1 summarizes the distinctions between PLMs without external knowledge integration and KEPLMs.

Despite the success of KEPLMs, two main problems still remain. (1) Most of the previous KEPLMs indiscriminately inject knowledge into PLMs, which can introduce noisy knowledge such as redundant or irrelevant information, potentially degrading model performance (Peters et al., 2019). These methods (Zhang et al., 2019; Liu et al., 2020; Wang et al., 2021b; Sun et al., 2020) inject corresponding knowledge triples or pre-trained knowledge embeddings into each entity in the context. However, some entities appear frequently in texts, leading to redundant knowledge injection. Irrelevant knowledge arises when some entities or their corresponding sub-graphs have little connection to the meanings of the underlying sentences; hence, they contribute minimally to the improvement of model performance. (2) Some methods modify model backbones with additional knowledge encoders, leading to inflexibility (Zhang et al., 2022b). Furthermore, optimizing these encoders can adversely affect the model's computational efficiency. Recently, some works (Sundararajan et al., 2017; Hao et al., 2021) use attribution to explain the mechanism of the Transformer. Most regard the self-attention layers as key-value pairs, while the study (Dai et al., 2022) views feed-forward networks (FFNs) as key-value memories and points out that some neurons in FFNs relate to knowledge expressions, motivating us to explore a similar spirit in KEPLMs.

Work done when Junbing Yan was doing an internship at Alibaba Group. Correspondence to Chengyu Wang and Wei Zhang.

LREC-COLING 2024, pages 16790–16801

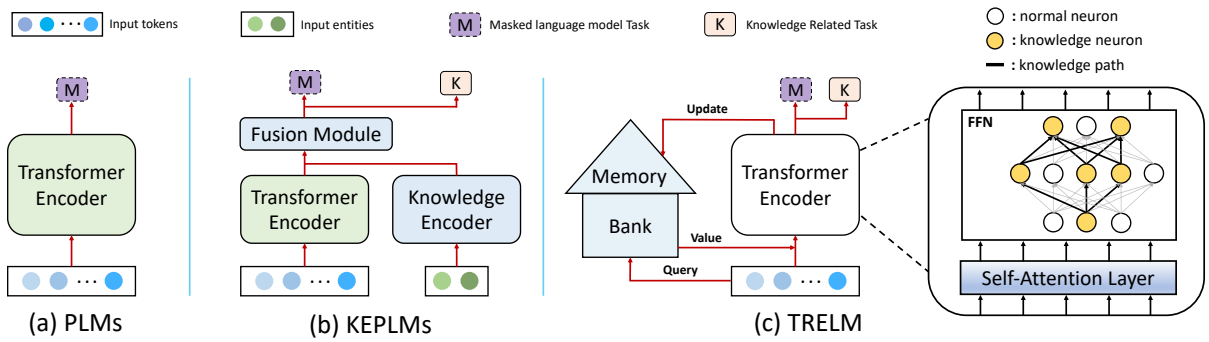

Figure 1: Comparison between TRELM and other models. (a) Plain PLMs usually utilize masked language modeling as the pre-training objective. (b) Some KEPLMs utilize external knowledge sources (e.g., KGs) and design knowledge-aware tasks which need additional knowledge encoders. (c) During pre-training, TRELM uses a BERT-style shared encoder and a knowledge-augmented memory bank to inject factual knowledge. Moreover, we only need to update partial FFN parameters in Transformer blocks with a dynamic knowledge routing method.

In this paper, we present our contributions in the form of our novel KEPLM training paradigm, namely TRELM, which enables the pre-training of more robust and efficient KEPLMs. To address the issue of excessive knowledge noise introduction, we propose identifying important entities as targets for knowledge injection. To facilitate the learning of improved representations, we construct a knowledge-augmented memory bank, which is vital in guiding the pre-training process and expediting convergence. Moreover, to optimize computational resource utilization, we introduce a technique called dynamic knowledge routing. This involves selective parameter updates within Transformer blocks. By identifying knowledge paths based on knowledge attribution, we enable partial updates of model parameters, focusing on the FFNs. Consequently, this results in a more efficient utilization of computing resources.

We conduct extensive experiments to verify the robustness and effectiveness of our TRELM framework over multiple NLP tasks. Our results show that TRELM outperforms strong baselines in knowledge-related tasks, including knowledge probing (LAMA) (Petroni et al., 2019), relation extraction, and entity typing. The pre-training time is also significantly reduced by over 50%. In summary, the contributions of this paper are as follows: [1]

- **New Pre-training Paradigm.** We introduce a more robust and efficient knowledge-enhanced pre-training paradigm (TRELM).
- **Knowledge-augmented Memory Bank.** We detect important entities in pre-training corpora and construct a knowledge-augmented memory bank, which guides the pre-training process and accelerates convergence.
- **Dynamic Knowledge Routing.** We propose a novel knowledge routing method that dynamically finds knowledge paths in FFNs and selectively updates model parameters.
- **Comprehensive Experiments.** We conduct extensive experiments and case studies to show the effectiveness and robustness of TRELM over various NLP tasks.

2. Related Work

In this section, we survey literature relevant to our study, encompassing three primary domains: KEPLMs, attribution methods in Transformer architectures, and the application of attribution to KEPLMs.

2.0.1. KEPLMs

KEPLMs incorporate external knowledge to enhance language understanding abilities of PLMs (Sun et al., 2019; Zhang et al., 2019; Peters et al., 2019; Xiong et al., 2020; Wang et al., 2021a; Liu et al., 2020; Wang et al., 2021b; Sun et al., 2020; Zhang et al., 2022b; Ye et al., 2022; Yu et al., 2022; Zhang et al., 2022a; Zhang et al., 2021b). For instance, ERNIE-Baidu (Sun et al., 2019) introduces entity and phrase level masking strategies to capture semantic information, while ERNIE-THU (Zhang et al., 2019) integrates entity embeddings into contextual representations using knowledge encoders. K-BERT (Liu et al., 2020) and CoLAKE (Sun et al., 2020) exploit knowledge graphs (KGs) to augment the language model with

[1] Source codes will be publicly available in the EasyNLP framework (Wang et al., 2022a). URL: https://github.com/alibaba/EasyNLP

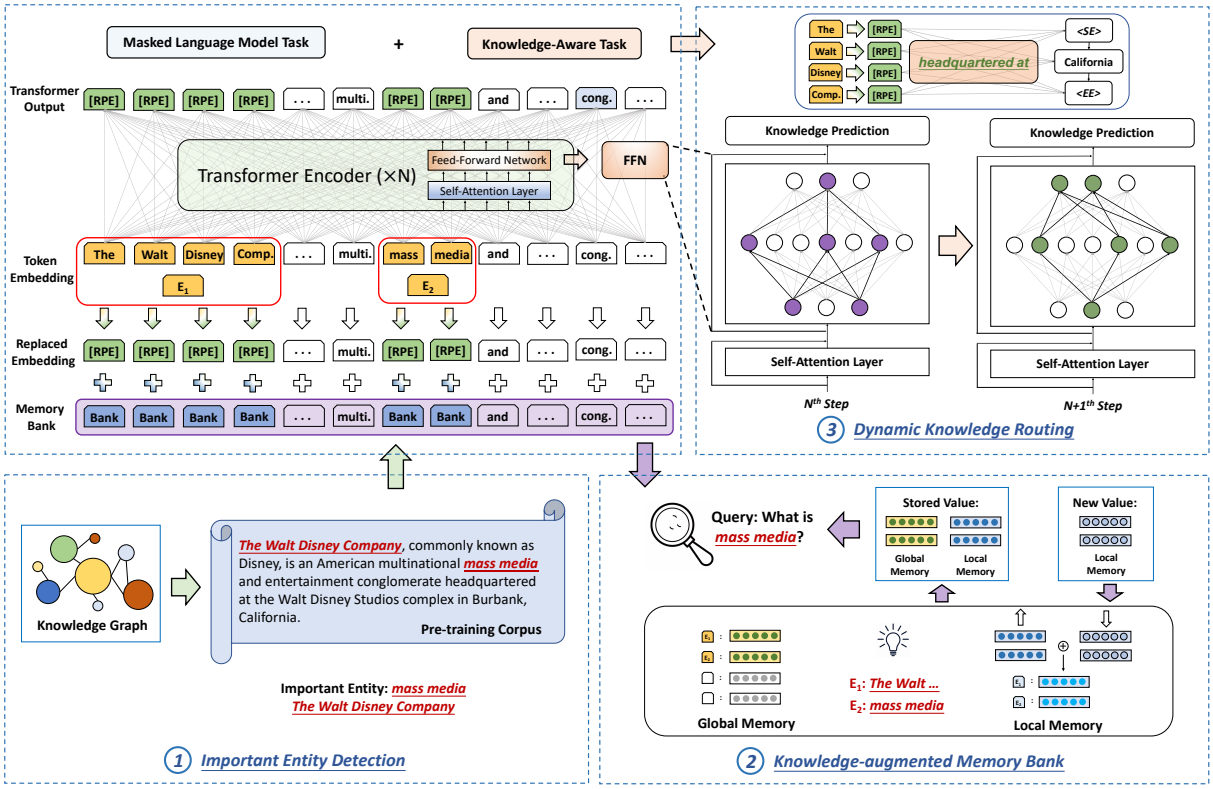

Figure 2: Model overview. (1) **Input:** Detecting important entities and long-tail words to reduce the knowledge noises. (2) **Knowledge-augmented Memory Bank:** Querying the important knowledge learned previously through a "cheat sheet" that contains semantic information of entities and words. (3) **Dynamic Knowledge Routing:** Finding the knowledge paths related to the knowledge-aware task, and selectively update the model's parameters.

graph structures, and DKPLM (Zhang et al., 2022b) employs shared encoders to unify texts and entities within a single semantic space. Despite these advancements, KEPLMs face ongoing challenges that limit their effectiveness and versatility. These challenges form the basis of our study and drive our exploration into novel methods for enhancing PLMs with external knowledge.

2.0.2. Attribution Methods in Transformers

Integrated gradients, a technique for attributing the model's output to its input features, has been increasingly adopted (Hao et al., 2021; Dai et al., 2022). For instance, Hao et al. (2021) applied integrated gradients to the self-attention mechanism, elucidating the importance of specific attention heads in the model's computations. More recent discussions by Wu et al. (2019) and Dong et al. (2021) have expanded the focus beyond self-attention, highlighting the significant role of FFNs within Transformers. Dai et al. (2022) employed integrated gradients to investigate the "knowledge neurons" in FFNs, providing insights into how these models process and store factual knowledge.

2.0.3. Attribution for KEPLMs

In the realm of KEPLMs, the challenge of filtering out knowledge noise has emerged as a critical concern. Several studies (Peters et al., 2019; Petroni et al., 2019; Cao et al., 2021; Sun et al., 2020; He et al., 2021; Zhang et al., 2022b; Wang et al., 2021b; Zhang et al., 2023) have demonstrated that the presence of knowledge noise can significantly impair model performance. Our research posits that the concept of knowledge paths, which are sequences of knowledge neurons within FFN layers of a Transformer, is instrumental to the effectiveness of KEPLMs.

3. TRELM: The Proposed Framework

We first state some basic notations. Denote an input token sequence as $x = (x_1, \cdots, x_i, \cdots, x_n)$, where n is the sequence length. The hidden representations of input tokens are denoted as $(h_1, h_2, \ldots, h_n)$ and $h_i \in \mathbb{R}^{d_1}$, where d_1 is the dimension of the representations. Furthermore, a knowledge graph is denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R})$. Here, $\mathcal{E}$ and $\mathcal{R}$ are the sets of entities and relation triples,

respectively. In the KG, a relational knowledge triple (e_h, r, e_t) comprises the head entity e_h, the relation r, and the tail entity e_t. The overall framework of TRELM is illustrated in Figure 2. We aim to address three key research questions:

- **RQ1**: How can we select useful positions and design more effective techniques for knowledge injection?
- **RQ2**: How can we ensure that our model retains the injected knowledge?
- **RQ3**: How can parameters of TRELM be efficiently updated during the pre-training knowledge learning process, while preserving downstream task performance?

3.1. Noise-aware Knowledge Injection

Important Entity Infusion. As shown by Petroni et al., 2019, Broscheit, 2019, Wang et al., 2020, and Cao et al., 2021, the semantics of high-frequency and common relation triples are already captured by plain PLMs. In this section, we aim to detect important entities for robust knowledge injection and knowledge noise reduction. In our work, selecting important entity positions in pre-training sentences for knowledge injection is vital. Inspired by Zhang et al. (2022b), we define the *Semantic Importance* (SI) score of the entity e in the sentence as $SI(e)$, indicating the semantic similarity between the representation of the original sentence and that of the sentence with e being replaced. We select our desired entities with high $SI(e)$ scores as target injection positions:

$$SI(e) = \frac{\|h_o\| \cdot \|h_{rep}\|}{h_o \cdot h_{rep}}. \quad (1)$$

Contrastive Knowledge Assessing. Knowing where to inject knowledge is insufficient, as knowledge constraints are only applied to the input layer. It is also necessary to verify whether the model truly acquires the knowledge. For model optimization, in addition to the Masked Language Modeling (MLM) task (Kenton and Toutanova, 2019), we propose *Contrastive Knowledge Assessing* (CKA) as an additional pre-training task.

The basic idea is that, given the representations of a head entity in the pre-training sentence and a relation at the input layer, the model needs to determine at the output layer whether it detects whether a given entity is the correct tail entity or not, and vice versa. Specifically, for the predicted i-th tokens of the tail entity h_d^i (Zhang et al., 2022b), we employ deep contrastive learning to encourage the model to capture the knowledge. Let $f(h_d^i, \cdot)$ be a matching function between h_d^i and a result token. The *token-level* CKA loss function is as follows:

$$\mathcal{L} = -\log \frac{\exp(f(h_d^i, y_i))}{\exp(f(h_d^i, y_i)) + \sum_{y_i' \sim Q(y_i)} \exp(f(h_d^i, y_i'))} \quad (2)$$

where y_i is the ground-truth token, and $y_i^{'}$ is a negative token sampled from a negative sampling function $Q(y_i)$. Hence, the total loss function of TRELM is:

$$\mathcal{L}_{\text{total}} = \theta \cdot \mathcal{L}_{\text{MLM}} + (1 - \theta) \cdot \mathcal{L}_{\text{CKA}} \quad (3)$$

where θ is the hyper-parameter, and $\mathcal{L}_{\text{CKA}}$ is the contrastive loss with respect to target entities.

3.2. Enhancing Representations with Knowledge-augmented Memory Bank

We have explored knowledge injection for important entities. Yet, since entities in the corpus typically follow a "long-tail" distribution (Wu et al., 2020; Zhang et al., 2022b), some representations can still be poorly optimized. Here, we further construct a *Knowledge-augmented Memory Bank* (KMB), which acts as a "cheat sheet" to ensure the model consistently captures important knowledge learned previously.

KMB Construction with Global and Local Memory Enhancement. Wu et al. (2020) discovered that learning representations for rare tokens during pre-training is challenging. It is reasonable to extend this hypothesis to knowledge-enhanced learning. However, their study focuses only on the local memory of infrequent tokens without considering the global memory of tokens. When encountering an important entity in a sentence, we can treat the contextual representations of its surrounding words as its "local memory." In detail, we construct a KMB $\mathcal{M}$. For an entity e present in both sentence x and $\mathcal{M}$, we denote the span boundary of e in x as (l, r), with l and r being the starting and ending positions, respectively. The "local memory" of e for x is defined as:

$$\mathcal{M}_{local}^{(e,x)} = \frac{1}{2k + r - l} \sum_{i=l-k}^{r+k} \mathbf{h}_i, \quad (4)$$

where $\mathbf{h}_i \in \mathbb{R}^{d_1}$ is the output at position i of the Transformer encoder, serving as the contextual representation of x. Here, k is half the window size and controls the number of surrounding tokens.

Since entity e may appear multiple times in the pre-training corpus, in $\mathcal{M}$, the "local memory" for entity e in KMB (denoted as $\mathcal{M}_{local}^{(e)}$) is updated using a moving average of every $\mathcal{M}_{local}^{(e,x)}$ that we obtain. We initialize $\mathcal{M}_{local}^{(e)}$ using the pre-trained embeddings of RoBERTa (Liu et al., 2019). Therefore, at any occurrence of entity e during pre-training,

its contextual information from all previous occurrences can be leveraged. We update $\mathcal{M}_{local}^{(e)}$ as:

$$\mathcal{M}_{local}^{(e)} \leftarrow (1-\gamma) \cdot \mathcal{M}_{local}^{(e)} + \gamma \cdot \mathcal{M}_{local}^{(e,x)} \quad (5)$$

where $\gamma \in (0,1)$ is the discount factor. Since the local memory contains localized information subject to isolation, we propose aggregating representations of e across multiple contexts as the "global memory". Let $\mathcal{T}^{(m)}$ be the collection of contexts involving entity e, i.e., $\mathcal{T}^{(m)} = \{\mathcal{T}_n | n \in \{1, \cdots, N\}, e \in \mathcal{T}_n\}$, and let $\mathbf{h}_{cls}$ be the output for the special <cls> classification token by the last Transformer layer. The "global memory" of entity e can be denoted as follows:

$$\mathcal{M}_{global}^{(e)} = \frac{1}{|\mathcal{T}^{(m)}|} \sum_{\mathcal{T}_n \in \mathcal{T}^{(m)}} \mathbf{h}_{cls}. \quad (6)$$

Leveraging KMB for Pre-training. We leverage the stored representations of entities in KMB as part of the input to the encoder. For any token sequence $x = \{x_1, \cdots, x_i, \cdots, x_n\}$, we first identify all important entities e appearing in x. Assuming that there are n important entities, they are denoted as $\{(e_i, l_i, r_i)\}_{i=1}^{n}$ where (l_i, r_i) are the boundaries of e_i in x at the i-th position respectively. If $l_i \leq p \leq r_i$, at position p, the input embeddings to the model are defined as follows:

$$\mathcal{I}_p = (1-\lambda) \cdot h_{e_i} + \frac{\lambda}{2} \cdot (\mathcal{M}_{local}^{(e_i)} + \mathcal{M}_{global}^{(e_i)}) \quad (7)$$

Otherwise, we have: $\mathcal{I}_p = \mathcal{E}_p$ where $\mathcal{E}_p$ is the token embedding at position p, h_{e_i} is the knowledge injection embedding for e_i, and λ is a hyper-parameter controlling the degree to which our KEPLM relies on KMB for contextual representations of important entities. We empirically set λ to 0.5 initially. To mitigate bias between pre-training and fine-tuning (which does not involve KMB), λ gradually decays to 0 towards the end of pre-training, i.e.:

$$\lambda_q = \frac{1}{\beta^q} \cdot \lambda, \quad q = 0, 1, 2, \cdots \quad (8)$$

where β is a hyper-parameter that controls the decay rate of λ, and q is the pre-training epoch.

3.3. Learning with Dynamic Knowledge Paths

After determining the model inputs and outputs, we proceed with the parameter optimization process. Building upon the hypothesis by Dai et al. (2022), which suggests that factual knowledge is stored in the FFN layers of Transformers, we introduce a *dynamic knowledge routing* algorithm to identify critical knowledge paths for TRELM updates during knowledge acquisition. Given an input sequence x, we define $\mathrm{P}_x(\hat{v}_i^{(l)})$ as the probability of producing the correct response according to the knowledge assessing objective:

$$\mathrm{P}_x(\hat{v}_i^{(l)}) = p(y^* | x, v_i^{(l)} = \hat{v}_i^{(l)}) \quad (9)$$

where p represents the Sampled SoftMax function; y^* is the correct response; $v_i^{(l)}$ is the i-th neuron in the l-th FFN layer; and $\hat{v}_i^{(l)}$ is a specific value of $v_i^{(l)}$. As $v_i^{(l)}$ varies from 0 to its upper bound $\overline{v}_i^{(l)}$, we calculate the neuron's attribution score by integrating the gradients of $P_x(\alpha v_i^{(l)})$:

$$\mathrm{Attr}(v_i^{(l)}) = \overline{v}_i^{(l)} \int_{\alpha=0}^{1} \frac{\partial \mathrm{P}_x(\alpha \overline{v}_i^{(l)})}{\partial v_i^{(l)}} d\alpha. \quad (10)$$

The attribution score, $Attr(v_i^{(l)})$, quantifies the impact of $v_i^{(l)}$ on the output probabilities using integrated gradients as α spans from 0 to 1. However, directly calculating the continuous integral is challenging; thus, we use the Riemann approximation:

$$\tilde{\mathrm{Attr}}(v_i^{(l)}) = \frac{\overline{v}_i^{(l)}}{m} \sum_{k=1}^{m} \frac{\partial \mathrm{P}_x(\frac{k}{m} \overline{v}_i^{(l)})}{\partial v_i^{(l)}} \quad (11)$$

where m is set to 20 based on empirical testing.

Neurons with high $\mathrm{Attr}(v_i^{(l)})$ scores are indicative of a strong association with the understanding of knowledge within FFN layers. We define the knowledge path in the $\mathcal{T}$-th FFN layer as the sequence:

$$(\mathcal{T}_{v_{input}^{(i)}} \rightarrow \mathcal{T}_{v_{inter}^{(j)}} \rightarrow \mathcal{T}_{v_{output}^{(k)}}) \quad (12)$$

where $\mathcal{T}_{v_{input}^{(i)}}$, $\mathcal{T}_{v_{inter}^{(j)}}$, and $\mathcal{T}_{v_{output}^{(k)}}$ represent the i-th, j-th, and k-th neurons associated with knowledge in the FFN's input, intermediate, and output layers, respectively. These connections are crucial to the factual knowledge present in KEPLM. By selectively updating the model parameters based on the gradients of these knowledge paths, we can significantly reduce the computational cost of pre-training. Our experiments confirm that this technique not only accelerates pre-training convergence but also improves the model's understanding capabilities.

Remarks. During pre-training, we efficiently identify knowledge paths for each batch in parallel by utilizing distinct knowledge decoding labels. Although the detection of knowledge paths adds some overhead, the reduction in back-propagation time during model pre-training far outweighs this initial cost. This streamlined approach not only enhances efficiency but also contributes to effectiveness in capturing relevant knowledge.

3.4. Summarization of Pre-training Process

We provide a summary of the entire pre-training procedure below.

Datasets	PLMs		KEPLMs						
	ELMo	RoBERTa	CoLAKE	KEPLER	DKPLM	KP-PLM	KALM	TRELM	△
Google-RE	2.2%	5.3%	9.5%	7.3%	10.8%	11.0%	10.2%	**11.5**%	+0.5%
UHN-Google-RE	2.3%	2.2%	4.9%	4.1%	5.4%	5.6%	5.2%	**5.9**%	+0.3%
T-REx	0.2%	24.7%	28.8%	24.6%	32.0%	32.3%	29.8%	**33.0**%	+0.7%
UHN-T-REx	0.2%	17.0%	20.4%	17.1%	22.9%	22.5%	22.6%	**23.3**%	+0.4%

Table 1: The performance on knowledge probing. △ represents the absolute improvements over the best results of existing KEPLMs compared to our model.

Datasets	BERT	TRELM$_{BERT}$	△
Google-RE	11.4%	**15.3**%	+3.9%
UHN-Google-RE	5.7%	**9.8**%	+4.1%
T-REx	32.5%	**36.7**%	+4.2%
UHN-T-REx	23.3%	**27.9**%	+4.6%

Table 2: The performance on knowledge probing based on BERT. △ represents the absolute improvements over BERT compared to TRELM.

Input. We identify important entities and long-tail words throughout the corpus. Entity embeddings are replaced with those generated by the KG embedding algorithm discussed in Section 3.1. Embeddings of important entities and long-tail words are then updated in the Knowledge-augmented Memory Bank (KMB).
Forward Pass. During each FFN layer, we calculate the attribution scores for the neurons. These scores allow us to evaluate the importance level of knowledge neurons and establish knowledge paths. Following a forward pass, KMB values are updated based on the output from the model's final Transformer layer.
Back Propagation. In the final step, we selectively update the parameters along the identified knowledge paths during back propagation, focusing the training on the most relevant aspects of the model's knowledge representation.

4. Experiments

In this section, we comprehensively evaluate the effectiveness of TRELM and compare it against state-of-the-art approaches.

4.1. Experimental Setup

Pre-training Data. For pre-training TRELM, we utilize the English Wikipedia dated 2020/03/01[2] as our data source. We align entities in the pre-training texts, recognized by entity linking tools such as TAGME (Ferragina and Scaiella, 2010), with the Wikidata5M (Wang et al., 2021b) knowledge graph. Wikidata5M provides a large-scale dataset that includes relation triples and entity description texts. Additional pre-processing and filtering steps are consistent with those used by ERNIE-THU (Zhang et al., 2019). As a result, our KG comprises 3,085,345 entities and 822 relation types, and we have prepared 26 million text sequences.
Baselines. We compare TRELM with the following state-of-the-art KEPLM approaches:

1. **ERNIE-THU** (Zhang et al., 2019): Integrates knowledge embeddings by introducing a new pre-training objective that aligns mentions with knowledge entities.
2. **KnowBERT** (Peters et al., 2019): Enhances language representations with structured knowledge through knowledge attention.
3. **KEPLER** (Wang et al., 2021b): Encodes entities alongside text within Transformer blocks to create a joint semantic space.
4. **CoLAKE** (Sun et al., 2020): Utilizes a knowledge graph and adjacency matrices to guide the information flow.
5. **DKPLM** (Zhang et al., 2022b): Detects long-tail entities and uses a shared encoder for the injection of knowledge triples.
6. **KP-PLM** (Wang et al., 2022b): Uses continuous prompts and introduces two knowledge-aware self-supervised tasks for pre-training.
7. **KALM** (Feng et al., 2022): Incorporates external knowledge into three levels of document contexts for language understanding.

4.2. Knowledge-aware Tasks

TRELM was evaluated on three knowledge-aware tasks: knowledge probing (in the zero-shot setting), relation extraction, and entity typing. Due to space constraints, the primary experiments utilized RoBERTa$_{BASE}$ as the underlying architecture. The results demonstrate TRELM's transferability to larger models.

[2]https://dumps.wikimedia.org/enwiki/

Model	MNLI (m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	AVG.
RoBERTa	87.5 / 87.3	91.9	**92.8**	94.8	63.6	91.2	90.2	78.7	86.4
KEPLER	87.2 / 86.5	91.5	92.4	94.4	62.3	89.4	89.3	70.8	84.9
CoLAKE	87.4 / 87.2	92.0	92.4	94.6	63.4	90.8	90.9	77.9	86.3
TRELM	**87.9** / **87.3**	**92.2**	92.6	**94.9**	**63.9**	**91.5**	**91.2**	**79.1**	**86.7**

Table 3: GLUE results on dev set. KEPLER, CoLAKE and TRELM are initialized with $\text{RoBERTa}_{\text{BASE}}$.

Model	Precision	Recall	F1
BERT	76.4±1.2	71.0±1.4	73.6±1.3
RoBERTa	77.4±1.8	73.6±1.7	75.4±1.8
ERNIE_{BERT}	78.4±1.9	72.9±1.7	75.6±1.9
$\text{ERNIE}_{RoBERTa}$	80.3±1.5	70.2±1.7	74.9±1.4
KnowBERT	77.9±1.3	71.2±1.5	74.4±1.3
KEPLER_{WiKi}	77.8±2.0	74.6±1.9	76.2±1.8
CoLAKE	77.0±1.6	75.7±1.7	76.4±1.5
DKPLM	79.2±1.3	75.9±1.2	77.5±1.2
KP-PLM	**80.8**±1.7	75.1±1.6	77.8±1.7
KALM	78.9±1.5	75.3±1.6	77.1±1.6
TRELM	80.2±1.3	**76.0**±1.4	**78.0**±1.2

Table 4: Model performance on Open Entity (%).

Model	Precision	Recall	F1
BERT	67.23±0.7	64.81±0.6	66.00±0.6
RoBERTa	70.80±0.5	69.60±0.6	70.20±0.5
ERNIE	70.01±0.8	66.14±0.7	68.09±0.7
KnowBERT	71.62±0.7	71.49±0.6	71.53±0.8
DKPLM	72.61±0.5	73.53±0.4	73.07±0.5
KP-PLM	72.60±0.8	73.70±0.7	73.15±0.7
KALM	72.52±0.8	73.38±0.9	72.95±0.8
TRELM	**72.89**±0.5	**73.84**±0.4	**73.36**±0.4

Table 5: Model performance on TACRED (%).

Knowledge Probing: The LAMA (Petroni et al., 2019) probes use cloze-style tasks (e.g., "Arroyo died at `[MASK]` in 1551.") to assess whether PLMs encapsulate factual knowledge. The LAMA-UHN (Pörner et al., 2019) subset presents a more challenging set of questions by removing samples that are easier to answer. TRELM's performance on these tasks was quantified using macro-averaged mean precision (P@1), which gauges the model's ability to retrieve correct facts accurately.

The results for the LAMA and LAMA-UHN tasks can be found in Table 1 and Table 2. BERT-based models were separated from RoBERTa-based ones due to the significantly smaller vocabulary size of BERT, as per insights from (Wang et al., 2021a). The primary findings are as follows: (1) Our model, built on RoBERTa-base, attains state-of-the-art results across four datasets. (2) To ensure a balanced comparison, TRELM was also trained on the BERT-base model. As displayed in Table 2, TRELM significantly surpasses BERT-base, with an average improvement of +4.2%, reinforcing that TRELM is an effective pre-training framework adaptable to various architectures.

Entity Typing: This task requires predicting the semantic type of a specified entity within a given context. To ensure a fair comparison, we adhere to the training settings used in (Zhang et al., 2022b) and evaluate TRELM on the Open Entity dataset (Choi et al., 2018). Consistent with prior studies, we report micro-averaged precision, recall, and F1 metrics. As Table 4 shows, KEPLMs generally outperform plain PLMs due to additional knowledge enhancements, with our TRELM model demonstrating superior performance through the integration of knowledge paths and memory.

Relation Extraction: The TRELM model was evaluated on the TACRED benchmark dataset (Zhang et al., 2017), which includes 42 types of semantic relations. We utilized both micro-averaged and macro-averaged metrics to assess performance. As shown in Table 5, TRELM achieved state-of-the-art performance, confirming the benefits of noise-aware knowledge injection and memory-augmented pre-training for Relation Extraction.

4.3. Language Understanding Tasks

TRELM was also tested on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). According to the results in Table 3, TRELM slightly outperforms RoBERTa and shows an average improvement of 0.4% over CoLAKE. Overall, the experiments validate TRELM's marked enhancement in knowledge-aware tasks and its competitive edge in general natural language understanding tasks.

4.4. Analysis of Pre-training Efficiency

Pre-training was conducted on a server equipped with eight NVIDIA Tesla A100-80G GPUs for both TRELM and DKPLM to ensure a fair comparison. The pre-training loss and F1 scores on Open Entity and TACRED, as illustrated in Figure 4 and Figure 5, demonstrate the efficiency of both models over time. As depicted in Figure 4, TRELM's loss converges more rapidly than that

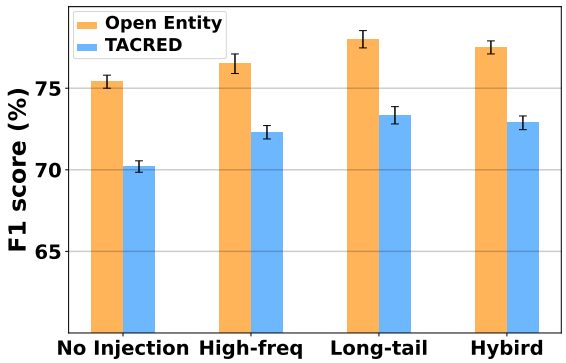

Figure 3: Injection method efficiency over Open Entity and TACRED.

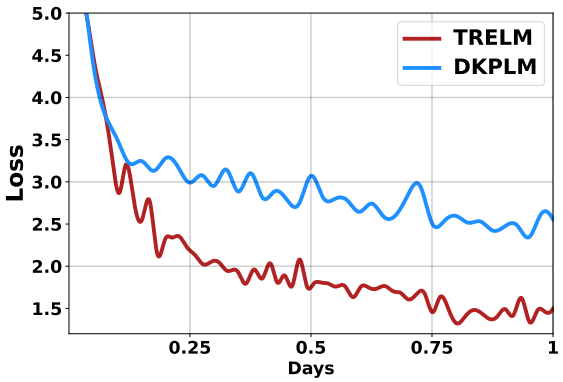

Figure 4: The curves of the pre-training loss.

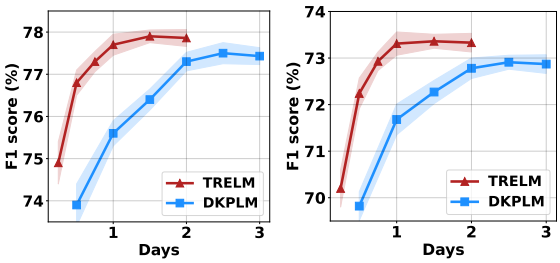

Figure 5: F1 score on Open Entity and TACRED for models trained under the same experiment setting.

of DKPLM, suggesting that the incorporation of a memory bank and dynamic knowledge routing contributes to faster model training. The loss curves for TRELM also exhibit greater smoothness, potentially reflecting the evolving quality of memory bank embeddings with continued training. By evaluating models using checkpoints saved at intervals of $\{0.25, 0.5, 0.75, 1, 1.5, 2\}$ days for TRELM and $\{0.5, 1, 1.5, 2, 2.5, 3\}$ days for DKPLM, we observe from Figure 5 that TRELM consistently outperforms DKPLM in terms of F1 scores. Notably, TRELM's performance within the first 0.75 days is comparable to that of DKPLM after 2 days, indicating that TRELM requires at least 50% less pre-training time to achieve similar results. In summary, TRELM reaches convergence in approximately one day, whereas DKPLM necessitates around two days, underscoring TRELM's greater pre-training efficiency.

4.5. Influence of Entities with Different Frequencies

We examined the impact of different knowledge injection strategies on TRELM, focusing on treatments involving only long-tail entities, only high-frequency entities, and a combination of the two. Utilizing the TACRED and Open Entity datasets, we measured the F1 score to assess the effectiveness of our noise-aware knowledge injection method. Figure 3 presents several key insights: (1) Injecting knowledge into long-tail entities yields better results than limiting it to high-frequency entities, suggesting a greater benefit in enriching representations for entities with sparse occurrences. (2) Superior performance can be achieved by selectively incorporating knowledge into specific subsets of entities, rather than indiscriminately targeting all available entities. (3) Our observations are consistent with the findings of Zhang et al., 2021a and Zhang et al., 2022b, which suggest that an overabundance of knowledge injection may detrimentally affect the model's effectiveness. These findings underscore the advantages of our method and the significance of strategic knowledge selection and injection in enhancing model performance.

4.6. Ablation Study

To elucidate the contributions of individual components, we conducted an ablation study and present the findings in Table 6. Specifically, the variant labeled "**-** Knowledge Injection" demonstrates a significant decline in the model's ability to comprehend language when noise-aware knowledge injection is removed, underscoring the importance of this feature for enhancing the base PLMs' performance. Similarly, the "**-** Knowledge Routing" results indicate not only that this component expedites the pre-training process but also that it makes a valuable contribution to the model's overall efficacy. These observations confirm that both knowledge injection and knowledge routing are integral to achieving the superior results.

4.7. Analysis of Each Module

In order to understand the individual contributions of each module, we carried out separate experiments to evaluate the specific impact of the Knowledge-augmented Memory Bank (KMB) and the Dynamic Knowledge Routing (DKR) on the pre-training efficiency of TRELM. We focused on quan-

Model	TACRED	Open Entity
TRELM	**73.34%**	**78.0%**
- Knowledge Injection	72.35%	76.8%
- Memory Bank	72.91%	77.7%
- Knowledge Routing	73.17%	77.6%

Table 6: Ablation study in terms of F1.

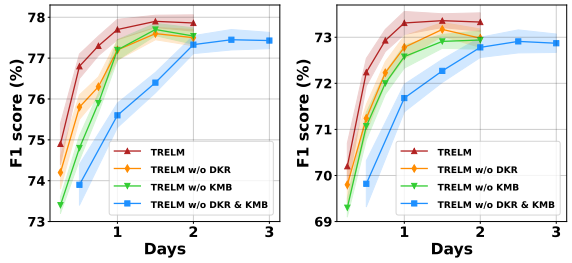

Figure 6: Efficiency of KMB and DKR over Open Entity and TACRED.

tifying the time saved by each module when used independently. Our analysis of the data illustrated in Figure 6 led to the following insights: (1) Both KMB and DKR enhance the convergence rate of TRELM in the pre-training phase. (2) KMB exhibits a more pronounced effect on expediting training in the early stages, while DKR's influence becomes increasingly significant over time, ultimately contributing to a greater overall efficiency. This trend may be attributed to an initial period where knowledge pathways are not yet fully established. As the model's capability to accurately assign knowledge improves, DKR's role in pinpointing precise knowledge paths intensifies, thereby boosting its contribution to training efficiency.

4.8. Hyper-parameter Analysis

We performed a detailed study on the Open Entity and TACRED datasets, focusing on three critical hyper-parameters: the balancing coefficients Θ for the contrastive knowledge-aware (CKA) task in Eq. 3, the decay rate β, and the half window size k. Each hyper-parameter was varied individually while keeping the others constant. As shown in Figure 7, performance initially improves with an increase in Θ, peaks at $\Theta = 0.5$, and then diminishes, suggesting an optimal trade-off between the CKA task and other learning objectives at this value. In Figure 8, a notable performance boost is observed as the half window size k rises from 4 to 16. However, this upward trend reverses when k is increased to 32, implying that an overly broad context window might introduce irrelevant information that hinders the model's learning. Referring to the same figure, the lowest performance is seen when $\beta = 1$, which corresponds to no decay and a consistent reliance on the memory bank across all pre-training. The

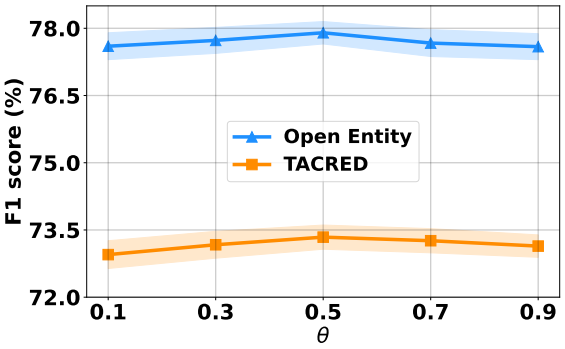

Figure 7: Hyper-parameter efficiency of θ over Open Entity and TACRED.

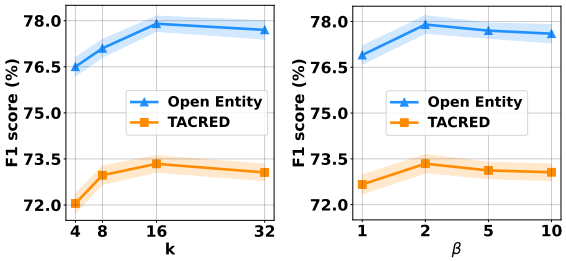

Figure 8: Hyper-parameter efficiency of k and β over Open Entity and TACRED.

model attains its highest performance at $\beta = 2$, but further increases in β lead to diminishing returns. This decline suggests that a more rapid decrease in the reliance on the memory bank limits the beneficial integration of knowledge, resulting in reduced model efficacy.

5. Conclusion

In this paper, we propose TRELM, a robust and efficient training paradigm for pre-training KEPLMs. TRELM introduces two innovative mechanisms designed to streamline the integration of knowledge into PLMs without requiring extra parameters: (1) a knowledge-augmented memory bank that prioritizes knowledge injection for important entities, and (2) a dynamic knowledge routing method that accelerates KEPLMs training and enhances language understanding by updating only the knowledge paths associated with factual knowledge. Our experiments demonstrate that TRELM achieves state-of-the-art performance on knowledge probing tasks and knowledge-aware language understanding tasks.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grant (No. 62072182) and Alibaba Group through Alibaba Research Intern Program.

6. Bibliographical References

Samuel Broscheit. 2019. Investigating entity knowledge in bert with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874.

Tyler Chang, Yifan Xu Xu, Weijian Xu, and Zhuowen Tu. 2021. Convolutions and self-attention: Re-interpreting relative positions in pre-trained language models. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 87–96.

Pedro Colon-Hernandez, Catherine Havasi, Jason B. Alonso, Matthew Huggins, and Cynthia Breazeal. 2021. Combining pre-trained language models and structured knowledge. *CoRR*, abs/2101.12294.

Leyang Cui, Sijie Cheng, Yu Wu, and Yue Zhang. 2021. On commonsense cues in bert for solving commonsense tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 683–693.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pages 2793–2803. PMLR.

Shangbin Feng, Zhaoxuan Tan, Wenqian Zhang, Zhenyu Lei, and Yulia Tsvetkov. 2022. KALM: knowledge-aware integration of local, document, and global contexts for long document understanding. *CoRR*, abs/2210.04105.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12963–12971.

Lei He, Suncong Zheng, Tao Yang, and Feng Zhang. 2021. Klmo: Knowledge graph enhanced pretrained language model with fine-grained relationships. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4536–4542.

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

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 the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

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

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models

as knowledge bases? In *EMNLP*, pages 2463–2473.

Nina Pörner, Ulli Waltinger, and Hinrich Schütze. 2019. BERT is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised QA. *CoRR*, abs/1911.03681.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *CoRR*, abs/2010.11967.

Chengyu Wang, Minghui Qiu, Taolin Zhang, Tingting Liu, Lei Li, Jianing Wang, Ming Wang, Jun Huang, and Wei Lin. 2022a. Easynlp: A comprehensive and easy-to-use toolkit for natural language processing. In *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022 - System Demonstrations, Abu Dhabi, UAE, December 7-11, 2022*, pages 22–29. Association for Computational Linguistics.

Jianing Wang, Wenkang Huang, Qiuhui Shi, Hongbin Wang, Minghui Qiu, Xiang Li, and Ming Gao. 2022b. Knowledge prompting in pre-trained language model for natural language understanding. *arXiv preprint arXiv:2210.08536*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuan-Jing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguistics*, 9:176–194.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Qiyu Wu, Chen Xing, Yatao Li, Guolin Ke, Di He, and Tie-Yan Liu. 2020. Taking notes on the fly helps BERT pre-training. *CoRR*, abs/2008.01466.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5412–5422.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Xiang Chen, Hui Chen, Feiyu Xiong, Xi Chen, and Huajun Chen. 2022. Ontology-enhanced prompt-tuning for few-shot learning. In *Proceedings of the ACM Web Conference 2022*, pages 778–787.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2022. Jaket: Joint pre-training of knowledge graph and language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11630–11638.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of*

the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 656–661.

Ningyu Zhang, Shumin Deng, Xu Cheng, Xi Chen, Yichi Zhang, Wei Zhang, and Huajun Chen. 2021a. Drop redundant, shrink irrelevant: Selective knowledge injection for language pretraining. In *IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4007–4014. ijcai.org.

Taolin Zhang, Zerui Cai, Chengyu Wang, Peng Li, Yang Li, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2021b. HORNET: enriching pre-trained language representations with heterogeneous knowledge sources. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2608–2617. ACM.

Taolin Zhang, Junwei Dong, Jianing Wang, Chengyu Wang, Ang Wang, Yinghui Liu, Jun Huang, Yong Li, and Xiaofeng He. 2022a. Revisiting and advancing chinese natural language understanding with accelerated heterogeneous knowledge pre-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: EMNLP 2022 - Industry Track, Abu Dhabi, UAE, December 7 - 11, 2022*, pages 560–570. Association for Computational Linguistics.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2022b. Dkplm: Decomposable knowledge-enhanced pre-trained language model for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11703–11711.

Taolin Zhang, Ruyao Xu, Chengyu Wang, Zhongjie Duan, Cen Chen, Minghui Qiu, Dawei Cheng, Xiaofeng He, and Weining Qian. 2023. Learning knowledge-enhanced contextual language representations for domain natural language understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15663–15676. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.