KEHRL: Learning Knowledge-Enhanced Language Representations with Hierarchical Reinforcement Learning

Dongyang Li[1,2], Taolin Zhang[2], Longtao Huang[2], Chengyu Wang[2], Xiaofeng He[1], Hui Xue[2]
[1]School of Computer Science and Technology, East China Normal University
[2]Alibaba Group
dongyangli0612@gmail.com, hexf@cs.ecnu.edu.cn
{zhangtaolin.ztl, kaiyang.hlt, chengyu.wcy, hui.xueh}@alibaba-inc.com

Abstract

Knowledge-enhanced pre-trained language models (KEPLMs) leverage relation triples from knowledge graphs (KGs) and integrate these external data sources into language models via self-supervised learning. Previous works treat knowledge enhancement as two independent operations, i.e., knowledge injection and knowledge integration. In this paper, we propose to learn **K**nowledge-**E**nhanced language representations with **H**ierarchical **R**einforcement **L**earning (KEHRL), which jointly addresses the problems of detecting positions for knowledge injection and integrating external knowledge into the model in order to avoid injecting inaccurate or irrelevant knowledge. Specifically, a high-level reinforcement learning (RL) agent utilizes both internal and prior knowledge to iteratively detect essential positions in texts for knowledge injection, which filters out less meaningful entities to avoid diverting the knowledge learning direction. Once the entity positions are selected, a relevant triple filtration module is triggered to perform low-level RL to dynamically refine the triples associated with polysemic entities through binary-valued actions. Experiments validate KEHRL's effectiveness in probing factual knowledge and enhancing the model's performance on various natural language understanding tasks.

Keywords: Knowledge-enhancement, Reinforcement learning, Pre-trained language model

1. Introduction

General pre-trained language models (PLMs) (Dong et al., 2019; Li et al., 2020; Bao et al., 2020) are pre-trained on various sources (Ma et al., 2020; Wu and He, 2019; Guu et al., 2020) and fine-tuned with specific data for diverse tasks, such as Information Extraction (Lee et al., 2022; Qin et al., 2021; Ma et al., 2021), Natural Language Inference (Qi et al., 2022; Saha et al., 2020), and Question Answering (Zhang et al., 2022a; Heo et al., 2022; Pappas and Androutsopoulos, 2021; Cheng et al., 2021).

To enhance context-aware representations, PLMs are equipped with additional knowledge collected from external resources in the forms of structured data such as relation triples from knowledge graphs (KGs) (Su et al., 2021) and unstructured description texts related to entities (Yu et al., 2022). This type of PLMs is often called knowledge-enhanced pre-trained language models (KEPLMs). Meanwhile, the recently emerged large language models (Zhao et al., 2023; Peng et al., 2023) also need external knowledge such as parametric knowledge (Luo et al., 2023) and retrieved knowledge (Pan et al., 2023) to augment themselves to alleviate hallucination (Zhang et al., 2023b; Liu et al., 2023).

According to previous research, KEPLMs (Wang et al., 2021b; Zhang et al., 2021b, 2022c) generally consist of two important modules, namely, knowledge injection and knowledge integration. (1) **Knowledge Injection**: it preprocesses the pre-training corpus into token-level input and chooses vital token positions (e.g., entities), preparing to inject the retrieved relevant relation triples from KGs at these positions. Recent research (Zhang et al., 2019; Peters et al., 2019; Liu et al., 2022) injects various types of knowledge into the positions of all entities indiscriminately. However, frequent and commonly used entities have already been learned sufficiently by PLMs, causing the redundant learning phenomenon and further inducing knowledge noise (Zhang et al., 2021a, 2022c). (2) **Knowledge Integration**: it aggregates the retrieved relation triples into the context-aware entity representations output by a plain PLM and learns new knowledge-enhanced representations to produce the final KEPLM. Existing works (Zhang et al., 2019; Lin et al., 2019; Peters et al., 2019) generally allocate different attention weights to each entity's triples to remove redundant knowledge. However, the soft distribution of the attention mechanism is not deterministic aggregation and inevitably introduces inaccurate knowledge such as irrelevant ambiguity triples (Peters et al., 2019; Zhang et al., 2019). As shown in Figure 1, since "CNBC" has a rela-

Work done when Dongyang Li was doing an internship at Alibaba Group. Dongyang Li and Taolin Zhang contributed equally to this work. Correspondence to Chengyu Wang and Xiaofeng He.

LREC-COLING 2024, pages 9693–9704
20-25 May, 2024.

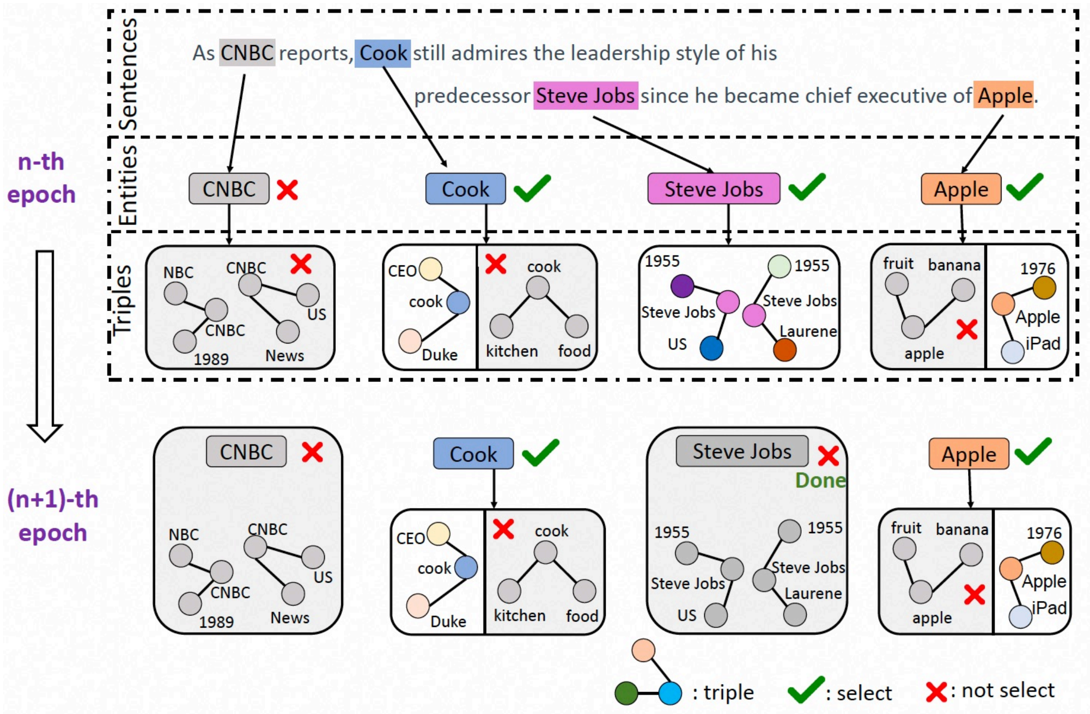

Figure 1: The dynamic selection process of entities and relation triples during pre-training.

tively weak contribution to the meaning of the whole sentence, we should attach less emphasis to it to avert error propagation to subsequent procedures. “Cook” and “Apple” are polysemic entities, and their irrelevant triples should be filtered out to avoid inaccurate knowledge. The well-learned entity “Steve Jobs” at the n-th epoch should not participate in the subsequent $(n+1)$-th epoch knowledge enhancement process to prevent duplicate learning (Zhang et al., 2021a, 2022c).

To tackle the problems mentioned above, we propose a new **K**nowledge-**E**nhanced language representation learning framework with **H**ierarchical **R**einforcement **L**earning (KEHRL) process to alleviate the error propagation problem, which jointly learns the positions of entities for knowledge injection and leverages relevant candidate relation triples dynamically at different levels for knowledge pre-training. Two new techniques are proposed and summarized below. **(1) Reinforced Entity Position Detection** combines the sentence's current representations and the prior knowledge as the state, leveraging high-level RL (Sutton and Barto, 1998) to detect the essential entity positions using the entity reward function derived based on the masked language modeling (MLM) task (Devlin et al., 2019). The final entity positions guide the model toward the prospective direction. When high-level RL's actions regarding entity positions are determined, low-level relevant triple selection will be triggered. **(2) Reinforced Triple Semantic Refinement** utilizes low-level RL to choose semantically valid relation triples of polysemic entities with binary-valued actions. We dynamically prune inaccurate and ambiguous relation triples according to the current state. The MLM task's token accuracy reward guides the model to adjust itself to calibrate the learning bias from irrelevant relation triples.

2. Pre-training Data Analysis

Selection of Entities We compare four different types of entity selection strategies for knowledge injection to observe changes in KEPLMs' performance, including (1) no entities selected, (2) all entities, (3) long-tail entities only, and (4) high-frequency entities only. We utilize BERT-base (Devlin et al., 2019) as the backbone to evaluate the performance. As Table 1 shows, we observe that knowledge injection into long-tail entities outperforms the high-frequency setting slightly, indicating that the models have already learned the factual knowledge well for these entities, where the relevant knowledge from KGs should be treated as redundant (Zhang et al., 2021a, 2022c). The knowledge injection setting of all entities has the lowest scores compared to the others. We suggest that the reason is that not all entities are helpful and different entities play different roles during the pre-training process for the enhancement of contextual semantics.

Knowledge Injection from Relation Triples Since PLMs have incorporated knowledge into model pa-

Tasks	Named Entity Recognition	Relation Extraction	Sentiment Analysis	Information Retrieval
Datasets	ACE2005	SemEval	SST-2	MARCO DOC DEV
Types ↓ Metrics →	F1	F1	ACC	MRR@100
No Entity	83.4	91.7	93.5	39.8
All Entities	82.5	89.6	92.5	38.3
Long-tail Entities	86.1	95.2	95.8	41.1
High-frequency Entities	84.8	93.9	94.0	40.4

Table 1: The performance of different entity selection types on four different tasks.

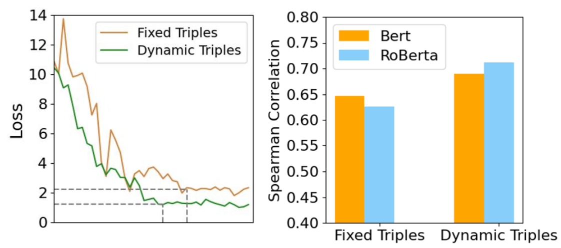

Figure 2: The comparison of different triple injection operations.

rameters (Su et al., 2021; Sun et al., 2020) through self-supervised learning, the remaining entities that are most difficult for the model to understand are polysemic entities (Zhang et al., 2023a), such as the ambiguous semantics of "apple" regarding Apple Inc. or a kind of fruit. Sentences injected with incorrect relation triples may divert them from their correct meanings (Liu et al., 2020). For example, when the important entity "Apple Inc." occurs in the pre-training sentence, it would be harmful if a relation triple about the apple as a fruit is injected, such as "<apple, subfamily_neighbour, pear>". We analyze two relation triple injection operations, including fixed triple injection and dynamic triple injection. Fixed triple injection leverages all correlated relation triples without any filtering techniques. Dynamic triple injection absorbs relation triples with attention-weighted values learned by self-supervised knowledge pre-training tasks automatically. We evaluate the average Spearman correlation score between the triple-injected training sentences and the original sentences. From Figure 2, we observe that dynamic triple injection has higher Spearman scores than fixed triple injection, validating that the dynamically triple-injected sentences contain less inaccurate semantics.

3. Model Architecture

In this section, we introduce our model components in detail. An overview is shown in Figure 3.

3.1. Model Notations

In the pre-training corpus, there are N_{total} sentences, and each sentence consists of certain tokens $S_i = (t_{i1}, t_{i2}, \cdots, t_{il_i})$. Each sentence includes N_i entities, and the sentence's entity collection can be denoted as $E_i = \{e_{i1}, e_{i2}, \cdots, e_{iN_i}\}$. The j-th entity of the i-th sentence is connected with M_{ij} triples in the KG; the entity's triple collection is denoted as $\{tri_{ij}^1, tri_{ij}^2, \cdots, tri_{ij}^{M_{ij}}\}$. We further denote d as the dimension of the model's hidden representations.

3.2. Reinforced Entity Position Detection

In this module, the policy learns to dynamically select the entity injection positions. The recognized entities in the sentence are regarded as the candidate pool. The more semantically important entities are selected, the higher the reward the strategy owns.

State: The representation of each entity is obtained in a knowledge combination process (See Sec. 3.4). The representation of the j-th entity in the i-th sentence e_{ij} is $H_{e_{ij}} \in \mathbb{R}^d$. We treat the concatenation of all the entity representations in the sample sentence as the state s_i^{high} of RL:

$$s_i^{\text{high}} = \{H_{e_{i1}} \mid\mid H_{e_{i2}} \mid\mid \cdots \mid\mid H_{e_{iN_i}}\} \quad (1)$$

where "||" denotes the operation of concatenation.

Policy: The policy of high-level RL is a probability distribution to decide which entity is more informative to the sentence. It leverages the current state to conduct related actions a_i^{high}. The policy $\pi_{\theta_{\text{high}}}$ is formulated as follows:

$$\pi_{\theta_{\text{high}}}(a_i^{\text{high}} | s_i^{\text{high}}) = P(a_i^{\text{high}} | s_i^{\text{high}}) \quad (2)$$

where θ_{high} represents the parameters of the policy.

Action: The action of high-level RL is to select the entities, and the action for the i-th sentence is

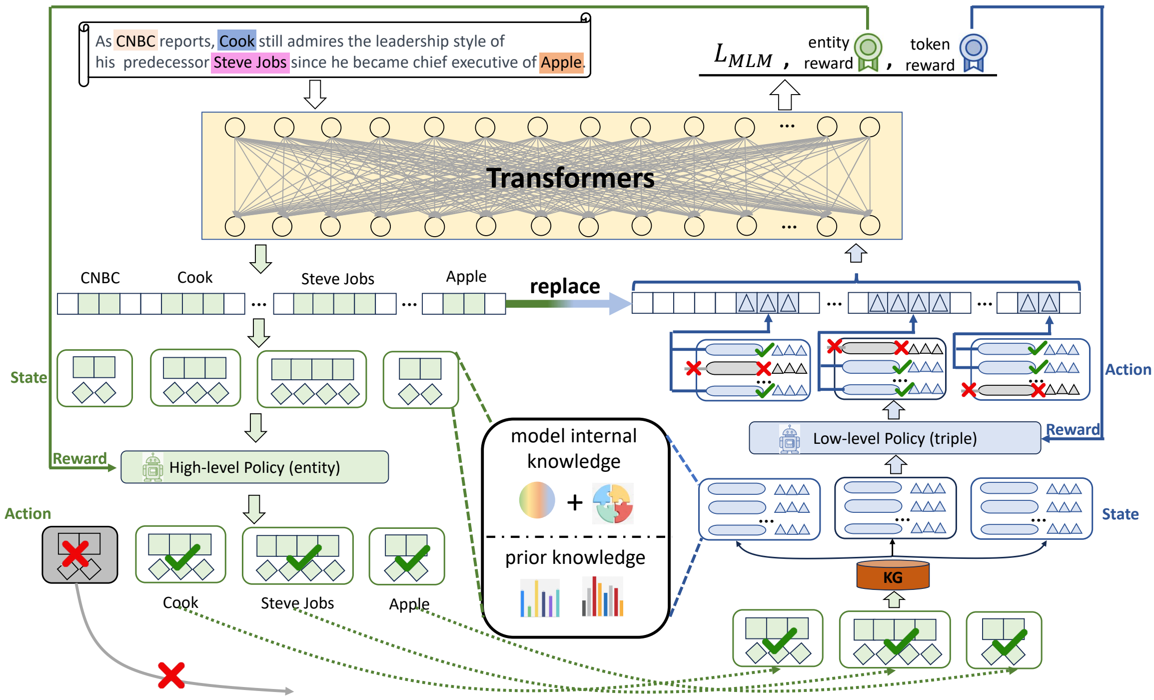

Figure 3: The model architecture of KEHRL. The green part represents Reinforced Entity Position Detection. The blue part represents Reinforced Triple Semantic Refinement.

a binary value vector:

$$a_i^{\text{high}} = \{0, 1, \cdots, 0\} \in \mathbb{R}^{N_i}, \\ a_i^{\text{high}} \sim \pi_{\theta_{\text{high}}}(a_i^{\text{high}} | s_i^{\text{high}}) \tag{3}$$

where “1” and “0” in the vector denote whether to select the entity or not. “$\sim$” means that the former is sampled from the distribution of the latter. Note that the low-level RL process for dynamic triple selection is only triggered when the high-level action for selecting entity injection positions is performed.

Reward: Our work treats the entity-grained task performance as the high-level reward instead of the intermediate reward. This depends on whether the correct entity is predicted in the MLM task. The reward can be formulated as follows:

$$r_{ij}^{\text{high}} = \begin{cases} 1 & \text{if } \hat{e}_{ij} = e_{ij}, \\ 0 & \text{if } \hat{e}_{ij} \neq e_{ij}. \end{cases} \tag{4}$$

where $\hat{e}_{ij}$ is the j-th entity predicted by the model for sentence S_i. The accumulated reward for the sentence is computed by $R_i^{\text{high}} = \sum_{j=1}^{|Mask_i^e|} r_{ij}^{\text{high}}$, where $|Mask_i^e|$ is the number of masked entities.

3.3. Reinforced Triple Semantic Refinement

At the low-level RL, the policy prompts the model to choose more accurate triples with respect to the selected entities during pre-training and filters out inaccurate triples, rather than relying on soft attention weights (Zhang et al., 2019; Lin et al., 2019; Peters et al., 2019).

State: We obtain the representation of each relation triple $H_{\text{tri}_{ij}^p} \in \mathbb{R}^d$ by knowledge combination (See Sec. 3.4). The concatenation of all triple representations retrieved based on the selected entities is treated as the state of low-level RL. Hence, the state is represented as:

$$s_{ij}^{\text{low}} = \{H_{\text{tri}_{ij}^1} \,||\, H_{\text{tri}_{ij}^2} \,||\, \cdots \,||\, H_{\text{tri}_{ij}^{M_{ij}}}\} \tag{5}$$

Policy: The policy of low-level RL determines a subset of related triples for a selected entity. It utilizes the representations of the selected entity's and all of the connected triples' to conduct the choosing operation. The policy is a distribution given the high-level action and the low-level states, i.e.,

$$\pi_{\theta_{\text{low}}}(a_{ij}^{\text{low}} | a_i^{\text{high}}; s_{ij}^{\text{low}}) = P(a_{ij}^{\text{low}} | a_i^{\text{high}}; s_{ij}^{\text{low}}) \tag{6}$$

Action: The action of low-level RL is selecting unambiguous relation triples and forcing the removal of inaccurate relation triples in the current training iteration. The action for the j-th entity in the i-th sentence is also a binary value vector to represent whether to choose the triple or not:

$$a_{ij}^{\text{low}} = \{0, 1, \cdots, 0\} \in \mathbb{R}^{M_{ij}}, \\ a_{ij}^{\text{low}} \sim \pi_{\theta_{\text{low}}}(a_{ij}^{\text{low}} | a_i^{\text{high}}; s_{ij}^{\text{low}}) \tag{7}$$

Reward: The final MLM task results at the token level are regarded as the low-level reward, similar to the high-level reward process. The low-level reward is the number of correctly predicted tokens in the MLM task, i.e.,

$$r_{iq}^{\text{low}} = \begin{cases} 1 & \text{if } \hat{t}_{iq} = t_{iq}, \\ 0 & \text{if } \hat{t}_{iq} \neq t_{iq}. \end{cases} \tag{8}$$

where $\hat{t}_{iq}$ is the q-th token predicted by the model for sentence S_i. The accumulated reward for the sentence is computed as $R_i^{\text{low}} = \sum_{q=1}^{|Mask_i^t|} r_{iq}^{\text{low}}$, where $|Mask_i^t|$ is the number of masked tokens. To inject the relation triples into PLMs, we utilize the triple representations to replace the related original entity representations at the entity positions.

3.4. Weighted Knowledge Combination

We leverage the weighted mix of two types of knowledge, including the model's internal knowledge (i.e., the context-aware representations) and prior knowledge, to further optimize the learning process.

Model's Internal Knowledge: Different granularities of text representations are combined to participate in the RL procedure. The entity's internal knowledge consists of the contexts and KGs. The contextual information is the mean-pooling representation $h_{e_{ij}} \in \mathbb{R}^d$, which is extracted from the sentence embedding between the entity start and end positions. The entity's KG information is the merged representations of all its connected triples, i.e.,

$$\begin{aligned} H_{e_{ij}}^{\text{mod}} &= h_{e_{ij}} + \sum_{p=1}^{M_{ij}} \alpha_{ip} h_{\text{tri}_{ij}^p}, \\ \alpha_{ip} &= \frac{h_{e_{ij}} \cdot h_{\text{tri}_{ij}^p}}{\sum_{p=1}^{M_{ij}} h_{e_{ij}} \cdot h_{\text{tri}_{ij}^p}}. \end{aligned} \tag{9}$$

The triple's internal knowledge consists of global-triple and sub-triple information. The global-triple is the specific pseudo sentence representation $h_{\text{tri}_{ij}^p} \in \mathbb{R}^d$. The pseudo sentence is the concatenation of the triples $\text{tri}_{ij}^p = \langle \text{head}, \text{rel}, \text{tail} \rangle$. The sub-triple is the merging of all triple components; we compute the subject, relation, and object representations respectively, and $k \in \{\text{head}, \text{rel}, \text{tail}\}$, i.e.,

$$\begin{aligned} H_{\text{tri}_{ij}^p}^{\text{mod}} &= h_{\text{tri}_{ij}^p} + \sum_{k=1}^{3} \beta_{ij}^{pk} h_{\text{tri}_{pk}^{ij}}, \\ \beta_{ij}^{pk} &= \frac{h_{\text{tri}_{ij}^p} \cdot h_{\text{tri}_{pk}^{ij}}}{\sum_{k=1}^{3} h_{\text{tri}_{ij}^p} \cdot h_{\text{tri}_{pk}^{ij}}}. \end{aligned} \tag{10}$$

Prior Knowledge: To calibrate the learning direction and avoid distorted forward steps, we consider prior knowledge as part of the enhanced knowledge components pre-processed before the model training stage. The entity's prior knowledge is the normalized appearance frequency of each entity relative to all entities in the training corpus. The triple's prior knowledge is the entity's connected (i.e., 1-hop, 2-hop, $\cdots$, k-hop) triples' normalized importance, calculated by semantic similarity:

$$\begin{aligned} H_{e_{ij}}^{\text{pri}} &= \text{softmax}\left(\frac{C_{e_{ij}}}{\sum_{i=1}^{N_{\text{total}}} \sum_{j=1}^{N_i} C_{e_{ij}}} \right), \\ H_{\text{tri}_{ij}^p}^{\text{pri}} &= \text{softmax}(\text{sim}(h_{\text{tri}_{ij}^p}, h_{S_i})). \end{aligned} \tag{11}$$

where $C_{e_{ij}}$ denotes the number of appearances of entity e_{ij}. "sim$(\cdot, \cdot)$" denotes the similarity using the cosine function. The sentence embedding $h_{S_i} \in \mathbb{R}^d$ is the sentence representation.

Next, we mix these two types of knowledge in a weighted operation:

$$\begin{aligned} H_{e_{ij}} &= \lambda H_{e_{ij}}^{\text{mod}} + (1 - \lambda) H_{e_{ij}}^{\text{pri}}, \\ H_{\text{tri}_{ij}^p} &= \lambda H_{\text{tri}_{ij}^p}^{\text{mod}} + (1 - \lambda) H_{\text{tri}_{ij}^p}^{\text{pri}}. \end{aligned} \tag{12}$$

where λ is the mixing degree controlling parameter.

3.5. Hierarchical Learning Strategy

Our total learning objectives are composed of three parts: the cross-entropy loss of the sentence MLM task, the high-level RL objective, and the low-level RL objective. The high-level RL objective is to maximize the expected accumulated rewards in the entity-grained MLM task:

$$J_{\theta_{\text{high}}} = \mathbb{E}_{s_i^{\text{high}}, a_i^{\text{high}}, r_{ij}^{\text{high}}} \sum_{j=1}^{|Mask_i^e|} r_{ij}^{\text{high}} \tag{13}$$

The low-level RL objective is to maximize the expected accumulated rewards in the token-level MLM task:

$$J_{\theta_{\text{low}}} = \mathbb{E}_{s_{ij}^{\text{low}}, a_{ij}^{\text{low}}, r_{iq}^{\text{low}}} \sum_{q=1}^{|Mask_i^t|} r_{iq}^{\text{low}} \tag{14}$$

Note that there are no discretized time steps in the episode (Sutton and Barto, 1998; Lapan, 2018). We model the episode as the training iteration. We treat the final task reward as the accumulated reward during the calculation stage. To enable the model to converge faster and exhibit smaller variance, we exploit policy gradient methods (Sutton et al., 1999) via the REINFORCE algorithm (Williams, 1992) with a baseline (Weaver and Tao, 2001) to optimize the RL objectives. The gradients for the high-level and low-level policy are as follows:

$$\begin{aligned}\nabla_{\theta_{\text{high}}} J_{\theta_{\text{high}}} &= \mathbb{E}_{s_i^{\text{high}}, a_i^{\text{high}}, r_{ij}^{\text{high}}}[(R_i^{\text{high}} - R_{\text{base}_i}^{\text{high}}) \\ &\qquad \nabla_{\theta_{\text{high}}} \pi_{\theta_{\text{high}}}(a_i^{\text{high}}|s_i^{\text{high}})] \\ \nabla_{\theta_{\text{low}}} J_{\theta_{\text{low}}} &= \mathbb{E}_{s_{ij}^{\text{low}}, a_{ij}^{\text{low}}, r_{iq}^{\text{low}}}[(R_i^{\text{low}} - R_{\text{base}_i}^{\text{low}}) \\ &\qquad \nabla_{\theta_{\text{low}}} \pi_{\theta_{\text{low}}}(a_{ij}^{\text{low}}|a_i^{\text{high}}; s_{ij}^{\text{low}})]\end{aligned} \tag{15}$$

Thus, the total training objective is:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \omega_1 \mathcal{L}_{\text{MLM}} + \omega_2 \mathcal{L}_{\text{high}} + \omega_3 \mathcal{L}_{\text{low}} \\ &= \omega_1 \mathcal{L}_{\text{MLM}} - \omega_2 J_{\theta_{\text{high}}} - \omega_3 J_{\theta_{\text{low}}}\end{aligned} \tag{16}$$

where $\omega_1, \omega_2, \omega_3$ are hyperparameters.

4. Experiments

4.1. Data and Baselines

(1) Pre-training Data. We fetch the pre-training samples from a wealth of knowledge resources, i.e., the English Wikipedia (2020/03/01). We obtain the detected entities' description text and related relation triples from WikiData5M (Wang et al., 2021b) using the entity linking tool TAGME (Ferragina and Scaiella, 2010). We follow ERNIE (Zhang et al., 2019) to complete the additional data processing stages. Finally, we obtain the pre-training data with 26 million samples, 3,085,345 entities, and 822 relation types.

(2) Downstream Data. Our work is evaluated by the LAMA benchmark[1] (Petroni et al., 2019). The four evaluation datasets of LAMA include approximately 2,550,000 sentences. Additionally, we introduce the Open Entity (Choi et al., 2018) with about 6,000 examples for the entity typing task, CoNLL2003 (Sang and Meulder, 2003) with about 22,000 examples for the named entity recognition task, and TACRED (Zhang et al., 2017) with 106,000 examples for the relation extraction task.

(3) Baselines. ERNIE (Zhang et al., 2019), **KnowBERT** (Peters et al., 2019), and **KALM** (Feng et al., 2023) inject the retrieved relevant entity embeddings into the model by an integrated entity linker. **KEPLER** (Wang et al., 2021b) and **DKPLM** (Zhang et al., 2022c) encode the entity's embedding and jointly optimize the model with knowledge embedding and MLM objectives. **GREASELM** (Zhang et al., 2022d) fuses the graph structure and language context representation to encourage them to perform well on textual narratives tasks. **KP-PLM** (Wang et al., 2022) is trained with multiple transformed knowledge sub-graph prompts.

[1] https://github.com/facebookresearch/LAMA/tree/main

4.2. Experiments Settings

During the pre-training stage, we utilize the RoBERTa base model as the backbone. We choose AdamW (Loshchilov and Hutter, 2017) as the optimizer with a learning rate of 5e-6 and a weight decay of 1e-5. The batch size is 184, and the model is trained for 5 epochs. The learning rate for the pre-training stage is set to 4e-5. The maximum hop number of the prior knowledge's entity connected triples k is set to 3. In KEHRL, we fix the number of entities per sentence to 5 and the number of triples per entity to 7. Sentences with fewer than 5 entities randomly select entities from the existing ones to fill the total entity count. This rule also applies to triples. The maximum length of the concatenated triples l_{tri} is 15. The ratio parameter λ controlling the internal and prior knowledge is 0.5. The proportion parameters ω of the total loss $\mathcal{L}_{\text{total}}$ are set to {0.3, 0.35, 0.35}. We run our pre-training stage on 8 NVIDIA A100 80G GPUs for 1 day.[2]

4.3. General Experimental Results

Zero-shot Knowledge Probing Tasks: We evaluate our model on the knowledge probing benchmark LAMA (Petroni et al., 2019) using the metric of macro-averaged mean precision (Mean P@1). As shown in Table 2, (1) KEHRL outperforms general PLMs and recent KEPLMs, owing to the RL-based knowledge selection. (2) The average improvements on T-REx related datasets are larger than those on Google-RE related datasets (3.5 vs. 0.5), demonstrating KEHRL's ability to probe factual knowledge in complex prediction scenarios. (3) The performance of KEHRL on four LAMA datasets is 2.1 points higher on average than the recent prevalent prompt-based model KP-PLM (Wang et al., 2022). We conjecture that KP-PLM focuses on prompt construction and pays less attention to filtering out inaccurate knowledge, indicating that our RL technique refines knowledge injection delicately and boosts performance.

Knowledge-intensive NLP Tasks via Fine-tuning: We validate our model on three knowledge-intensive tasks to verify its knowledge learning quality. The details are as follows:

In the entity linking task, we observe that (1) KEHRL achieves the best results in terms of the F1 metric and outperforms the strongest baseline KALM (Feng et al., 2023) by 3.8 points (from 78.1 to 81.9) because KALM integrates abundant knowledge without meticulous detection of inaccurate samples. (2) Compared with the baselines, KEHRL

[2] The source code and data can be available at https://github.com/MatNLP/KEHRL

Datasets↓ Models→	PLMs		KEPLMs						
	ELMo	RoBERTa	KEPLER	GREASELM	DKPLM	KP-PLM	KALM	KEHRL	Δ
Google-RE	2.2	5.3	7.3	10.6	10.8	11.0	10.9	**11.6**	+0.6
UHN-Google-RE	2.3	2.2	4.1	5.0	5.4	5.6	5.4	**5.9**	+0.3
T-REx	0.2	24.7	24.6	26.8	32.0	32.3	31.1	**34.9**	+2.6
UHN-T-REx	0.2	17.0	17.1	22.7	22.9	22.5	23.1	**27.5**	+4.4

Table 2: Experimental results of KEHRL and baselines on the LAMA benchmark in terms of Mean P@1 metric (%). The t-tests demonstrate the improvements of KEHRL are statistically significant with p <0.05.

Models↓ Datasets→	Open Entity			CoNLL2003			TACRED		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BERT	76.4	72.0	73.6	91.6	93	92.4	67.2	64.8	66.0
RoBERTa	77.4	73.6	75.4	90.9	94.4	92.6	70.8	69.6	70.2
ERNIE$_{BERT}$	78.4	72.9	75.6	89.5	94.2	91.8	70.0	66.1	68.1
KnowBERT$_{BERT}$	77.9	71.2	74.4	91.2	92.8	92.0	71.6	71.5	71.5
GREASELM	80.1	74.8	77.4	91.6	94.0	92.8	73.9	73.2	73.5
DKPLM	79.2	**75.9**	77.5	92.5	93.7	93.1	72.6	73.5	73.1
KP-PLM	80.8	75.1	77.8	92.7	93.9	93.3	72.6	73.7	73.2
KALM	82.5	75.2	78.7	92.3	93.5	92.9	74.7	73.8	74.2
KEHRL	**89.3**	75.6	**81.9**$_{(\pm 0.7)}$	**92.8**	**94.6**	**93.7**$_{(\pm 0.3)}$	**78.0**	**74.1**	**76.0**$_{(\pm 0.4)}$

Table 3: The experimental results (%) on downstream knowledge-intensive tasks.

attains the highest precision score of 89.3. These results confirm that the tailored knowledge injection mechanism of KEHRL effectively incorporates knowledge to enhance performance. In the named entity recognition task, KEHRL achieves the best F1 score of 93.7 (+0.4) on this knowledge-intensive task, illustrating that our model has accurate representational ability in entity-aware scenarios, assisted by the knowledge refinement of RL. In the relation extraction (RE) task, we fine-tune our model using the training set and test the model's relation extraction ability. The model yields the highest scores across all three metrics (+3.3 in Precision, +0.3 in Recall, and +1.8 in F1), further indicating KEHRL's accurate representations of entities and relations, due to the judicious knowledge injection.

5. Detailed Analysis of KEHRL

5.1. Ablation Study

To evaluate the effectiveness of each important module in our model, we conduct an ablation study on the Open Entity (Choi et al., 2018) and TACRED (Zhang et al., 2017) datasets. The results are as follows:

1. When Reinforced Triple Semantic Refinement is removed, all related triples are injected into the model without meticulous refining, causing a performance drop of approximately 3.5 and 0.8 in F1 on the two datasets, respectively.

2. Without Reinforced Entity Position Detection, knowledge is injected at all entity positions without essential position detection, which may introduce irrelevant knowledge, leading to a significant decline in model performance (81.9 → 74.2, and 76.0 → 71.8 in F1).

3. Removing the weighted mixed knowledge and retaining only internal knowledge while ignoring prior knowledge results in a decrease of 1.4 and 0.5 points in F1.

Methods	Open Entity	TACRED
KEHRL	81.9	76.0
- Trip. Sema. Refi.	78.4	75.2
- Enti. Posi. Dete.	74.2	71.8
- Weig. Mixed Knowledge	80.5	75.5

Table 4: Ablation study of KEHRL on Open Entity and TACRED in terms of F1 metric. "-" means removing the module.

The ablation experiments indicate that each module contributes significantly to the model's performance on these tasks.

5.2. The Influence of RL

To probe the value of our customized knowledge injection operation, we compare our RL-refined model with a naive model that injects knowledge at all entity positions on QA tasks, including MS MARCO (Nguyen et al., 2016) and MQ2008 (Qin and Liu, 2013). We randomly sample three queries' top-10 related passages from MS MARCO and three

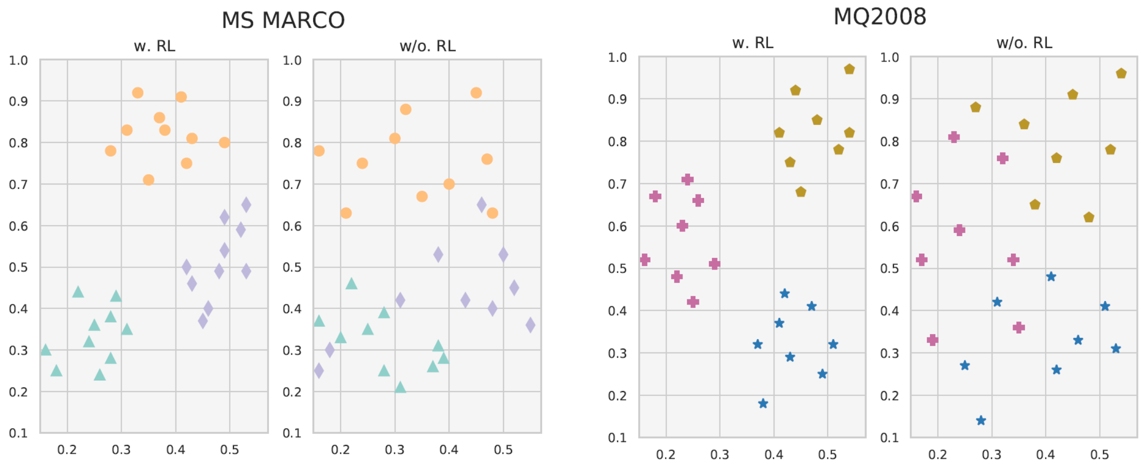

Figure 4: The influence of Reinforcement Learning on MS MARCO and MQ2008.

Methods	Open Entity	TACRED
KEHRL	81.9	76.0
No Prior	80.5	75.5
Element Type Prior	79.3	73.4
Model Structure Prior	82.0	76.3
Data Augmentation Prior	81.7	75.9

Table 5: The comparison of different prior knowledge combination methods.

queries' 8 related passages from MQ2008[3], then feed these passages to the two different models.

In Figure 4, we visualize the passage representations after t-SNE (van der Maaten and Hinton, 2008) dimensional reduction. The first and third figures represent passages from our meticulously RL-refined model, while the second and fourth figures represent passages from the model with naive knowledge injection at all entity positions. The closely clustered representations of our RL model indicate its capability to select accurate and informative knowledge. In contrast, the sparse distribution of the naive model's representations suggests less accurate knowledge selection.

A case study further highlights our model's effectiveness. As shown in Figure 5, KEHRL correctly identifies meaningful entities and relation triples, allocating less attention to less significant entities such as "CNBC" and its related triples, thereby demonstrating the precision of our RL-based knowledge injection.

5.3. The Influence of Weighted Mixed Knowledge

We explore the impact of different prior knowledge strategies on our model:

- **Element Type Prior:** Projects entity and relationship information into vectors for integration into the model.
- **Model Structure Prior:** Incorporates an additional model to preprocess training samples and generate representations for entities and triples.
- **Data Augmentation Prior:** Applies Easy Data Augmentation (EDA) techniques (Wei and Zou, 2019) to samples and pseudo triple sentences, considering the augmented representations as prior knowledge.

According to the results presented in Table 5, both the No Prior and Element Type Prior strategies yield lower scores on the two datasets when compared to KEHRL. The Model Structure Prior achieves the best performance, surpassing that of KEHRL. However, this approach introduces an additional computational burden due to the extra model required. The Data Augmentation Prior performs marginally below KEHRL and entails increased computational costs associated with the data augmentation process. Consequently, our approach of leveraging linguistic prior knowledge proves to be the most cost-effective alternative.

6. Related Works

6.1. Knowledge-Enhanced Pre-trained Language Models (KEPLMs)

KEPLMs leverage external knowledge from Knowledge Graphs (KGs) to enhance semantic representation capabilities. They can be categorized based on the type of knowledge used:

1. **Structured Knowledge:** Works by Zhang et al. (2022b, 2023c); Su et al. (2021); Sun

[3] In MQ2008, each query only contains 8 related passages.

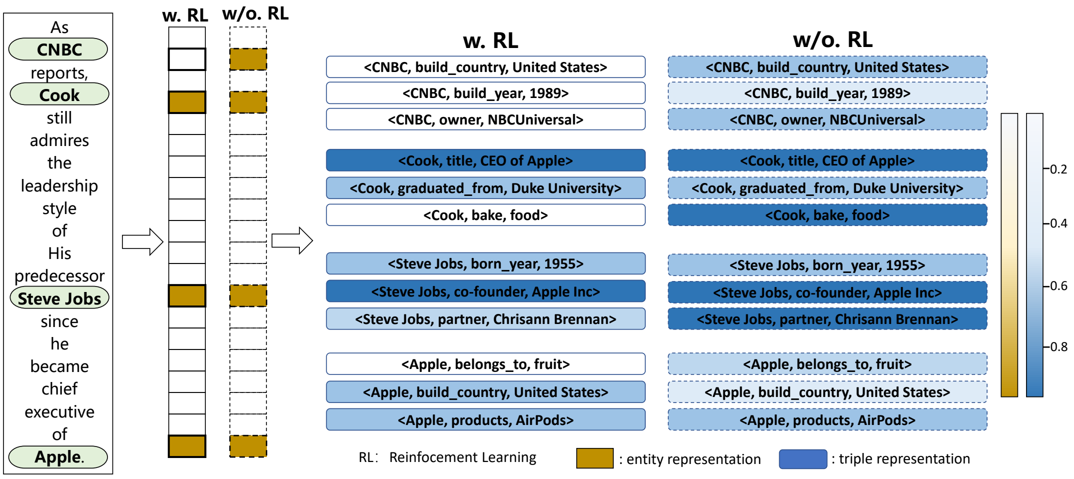

Figure 5: The heat map of the entity and triple representations of KEHRL and the model without RL. The solid line produced by KEHRL and the dotted line is the model without RL.

et al. (2020); Ji et al. (2020); Lin et al. (2019) augment models with sub-graphs from KGs by collecting multi-hop triples, learning nuanced semantics through graph neural networks and attention mechanisms.

2. **Unstructured Knowledge:** Yu et al. (2022); Chen et al. (2022) employ dictionary descriptions of sentence components to bolster the models' information retention. For instance, RAG (Lewis et al., 2020) retrieves top-k related text documents or chunks using the K-NN algorithm to enrich the training corpus.

3. **Heterogeneous Knowledge:** Qin et al. (2021) incorporate both entity and relation representations in the neighboring space for enhancement. K-Adapter (Wang et al., 2021a) integrates contextual relation semantics of entities into the model through a pluggable training strategy.

However, previous approaches generally treat knowledge integration as two separate processes, not considering entity selection and triple refinement jointly.

6.2. Hierarchical Reinforcement Learning (HRL)

Hierarchical Reinforcement Learning (HRL) breaks down complex problems into manageable subtasks, each addressed independently.

1. **Top-down HRL:** This approach uses a high-level policy to determine low-level settings. For instance, Takanobu et al. (2019) divided the relation extraction task into high-level relation detection and low-level entity extraction. In medical applications, Zhong et al. (2022) designed a master model to activate symptom checkers and disease classifiers. Rohmatillah and Chien (2023) established a high-level domain at the start of a dialogue, with sub-polices controlling the subsequent conversation.

2. **Bottom-up HRL:** This type focuses on low-level policies aiding high-level policy learning. HRL-Rec (Xie et al., 2021) has a low-level agent for channel selection, which guides high-level item recommendations. VISA (Jonsson and Barto, 2006) decomposes the value function and employs a Dynamic Bayesian Network to model relationships.

7. Conclusion

In this paper, we introduce KEHRL, a pre-training framework utilizing Hierarchical Reinforcement Learning for natural language understanding. The Reinforced Entity Position Detection module selects knowledge injection positions intelligently, avoiding less meaningful ones. The Reinforced Triple Semantic Refinement filters out inaccuracies and focuses on relevant triples linked to the chosen entities from the preceding module. Extensive experiments verify the effectiveness of KEHRL on factual knowledge based task and knowledge-intensive language tasks.

Acknowledgements

We would like to thank anonymous reviewers for their valuable comments. This work was supported in part by National Key R&D Program of China (No. 2022ZD0120302) and Alibaba Group through Alibaba Research Intern Program.

Bibliographical References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML*, volume 119, pages 642–652.

Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang. 2022. Dictbert: Dictionary description knowledge enhanced language model pre-training via contrastive learning. In *IJCAI*, pages 4086–4092.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. Unitedqa: A hybrid approach for open domain question answering. In *ACL*, pages 3080–3090.

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

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*, pages 13042–13054.

Shangbin Feng, Zhaoxuan Tan, Wenqian Zhang, Zhenyu Lei, and Yulia Tsvetkov. 2023. KALM: knowledge-aware integration of local, document, and global contexts for long document understanding.

Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, pages 1625–1628.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *ICML*, volume 119, pages 3929–3938.

Yu-Jung Heo, Eun-Sol Kim, Woo Suk Choi, and Byoung-Tak Zhang. 2022. Hypergraph transformer: Weakly-supervised multi-hop reasoning for knowledge-based visual question answering. In *ACL*, pages 373–390.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *EMNLP*, pages 725–736.

Anders Jonsson and Andrew G. Barto. 2006. Causal graph based decomposition of factored mdps. *J. Mach. Learn. Res.*, 7:2259–2301.

Maxim Lapan. 2018. *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd.

Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. Formnet: Structural encoding beyond sequential modeling in form document information extraction. In *ACL*, pages 3735–3754.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *EMNLP*, pages 9119–9130.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP*, pages 2829–2839.

Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen. 2023. RETA-LLM: A retrieval-augmented large language model toolkit. *CoRR*, abs/2306.05212.

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 *AAAI*, pages 2901–2908.

Xiaokai Liu, Feng Zhao, Xiangyu Gui, and Hai Jin. 2022. Lekan: Extracting long-tail relations via layer-enhanced knowledge-aggregation networks. In *DASFAA*, volume 13245, pages 122–136.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and

Daxin Jiang. 2023. Augmented large language models with parametric knowledge guiding. abs/2305.04757.

Ruotian Ma, Tao Gui, Linyang Li, Qi Zhang, Xuanjing Huang, and Yaqian Zhou. 2021. SENT: sentence-level distant relation extraction via negative training. In *ACL*, pages 6201–6213.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Charbert: Character-aware pre-trained language model. In *COLING*, pages 39–50.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *NIPS*, volume 1773 of *CEUR Workshop Proceedings*.

Junting Pan, Ziyi Lin, Yuying Ge, Xiatian Zhu, Renrui Zhang, Yi Wang, Yu Qiao, and Hongsheng Li. 2023. Retrieving-to-answer: Zero-shot video question answering with frozen large language models. *CoRR*, abs/2306.11732.

Dimitris Pappas and Ion Androutsopoulos. 2021. A neural model for joint document and snippet ranking in question answering for large document collections. In *ACL*, pages 3896–3907.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. abs/2302.12813.

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

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.

Kunxun Qi, Hai Wan, Jianfeng Du, and Haolan Chen. 2022. Enhancing cross-lingual natural language inference by prompt-learning from cross-lingual templates. In *ACL*, pages 1910–1923.

Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: improving entity and relation understanding for pre-trained language models via contrastive learning. In *ACL*, pages 3350–3363.

Mahdin Rohmatillah and Jen-Tzung Chien. 2023. Hierarchical reinforcement learning with guidance for multi-domain dialogue policy. *IEEE ACM Trans*, 31:748–761.

Swarnadeep Saha, Yixin Nie, and Mohit Bansal. 2020. Conjnli: Natural language inference over conjunctive sentences. In *EMNLP*, pages 8240–8252.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *NAACL*, pages 142–147. ACL.

Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. *AI Open*, 2:127–134.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *COLING*, pages 3660–3670.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, pages 1057–1063.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI*, pages 7072–7079.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. pages 2579–2605.

Jianing Wang, Wenkang Huang, Minghui Qiu, Qiuhui Shi, Hongbin Wang, Xiang Li, and Ming Gao. 2022. Knowledge prompting in pre-trained language model for natural language understanding. In *EMNLP*, pages 3164–3177.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of ACL*, 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. *TACL*, 9:176–194.

Lex Weaver and Nigel Tao. 2001. The optimal reward baseline for gradient-based reinforcement learning. In *UAI*, pages 538–545.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, pages 6381–6387.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *CIKM*, pages 2361–2364.

Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Hierarchical reinforcement learning for integrated recommendation. In *AAAI*, pages 4521–4528. AAAI Press.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022. Dict-bert: Enhancing language model pre-training with dictionary. In *Findings of the ACL*, pages 1907–1918.

Denghui Zhang, Yanchi Liu, Zixuan Yuan, Yanjie Fu, Haifeng Chen, and Hui Xiong. 2023a. Multi-faceted knowledge-driven pre-training for product representation learning. *TKDE*, 35(7):7239–7250.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022a. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *ACL*, pages 5773–5784.

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*, pages 4007–4014.

Shuo Zhang, Liangming Pan, Junzhou Zhao, and William Yang Wang. 2023b. Mitigating language model hallucination with interactive question-knowledge alignment. *CoRR*, abs/2305.13669.

Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. 2021b. Smedbert: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining. In *ACL*, pages 5882–5893.

Taolin Zhang, Junwei Dong, Jianing Wang, Chengyu Wang, Ang Wang, Yinghui Liu, Jun Huang, Yong Li, and Xiaofeng He. 2022b. Revisiting and advancing chinese natural language understanding with accelerated heterogeneous knowledge pre-training. In *EMNLP*, pages 560–570.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2022c. DKPLM: decomposable knowledge-enhanced pre-trained language model for natural language understanding. In *AAAI*, pages 11703–11711.

Taolin Zhang, Ruyao Xu, Chengyu Wang, Zhongjie Duan, Cen Chen, Minghui Qiu, Dawei Cheng, Xiaofeng He, and Weining Qian. 2023c. Learning knowledge-enhanced contextual language representations for domain natural language understanding. In *EMNLP*, pages 15663–15676.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022d. Greaselm: Graph reasoning enhanced language models for question answering.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *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 *ACL*, pages 1441–1451.

Ruochen Zhao, Xingxuan Li, Shafiq R. Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. abs/2305.03268.

Cheng Zhong, Kangenbei Liao, Wei Chen, Qianlong Liu, Baolin Peng, Xuanjing Huang, Jiajie Peng, and Zhongyu Wei. 2022. Hierarchical reinforcement learning for automatic disease diagnosis. *Bioinform.*, 38(16):3995–4001.