Improving Pre-trained Language Models with Knowledge Enhancement and Filtering Framework

Qi Zhao[1], **Qi Song**[1*], **Tian Xie**[1], **Haiyue Zhang**[2], **Hongyu Yang**[1], **Xiangyang Li**[1]
[1]University of Science and Technology of China, Hefei, Anhui, China
[2]Columbia University, New York, NY, USA
{zq2021, xie_tian, hongyuyang}@mail.ustc.edu.cn
hz2995@columbia.edu, {qisong09, xiangyangli}@ustc.edu.cn

Abstract

Pre-trained language models (PLMs) are widely used in NLP but struggle with capturing entity knowledge. To address this, knowledge enhancement techniques have been proposed. However, existing methods rely heavily on external knowledge bases embedding and often introduce noisy entity representations. In this work, we propose a novel **K**nowledge **E**nhancement **F**iltering **F**ramework named KEFF, which contains both knowledge enhancement and knowledge enhancement filtering modules for PLM. We find that there are certain redundant bits in the embedding space of PLMs. Building on this insight, we implement knowledge-enhanced mapping of redundant bit values in entity span tokens. In order to solve the knowledge enhancement problem of existing methods that introduce noisy entity representation knowledge, we further propose a novel knowledge enhancement filter based on our knowledge enhancement method. Finally, experiments on four knowledge-driven NLP tasks show that our method effectively improves the ability of PLMs on downstream tasks. Compared to state-of-the-art approachs, our method achieves the highest F1-score and accuracy, while reducing the computational cost by 1.7-2.5x.

1 Introduction

Pre-trained language models (PLMs), such as BERT (Devlin et al., 2018), have become central to many recent NLP applications. Research indicates that these models have a certain degree of factual knowledge (Petroni et al., 2019). However, PLMs perform poorly when facing some entity-related downstream tasks such as entity recognition (Li et al., 2020), entity relationship classification (Li et al., 2022), and entity typing (Choi et al., 2018; Ding et al., 2021). In recent years, the field of knowledge-enhanced pre-trained language modeling has seen rapid advancements, with numerous methods incorporating external knowledge bases like knowledge graphs (KGs (Peng et al., 2023)) to effectively enhance the entity-related knowledge capabilities of PLMs (Yang et al., 2021; Hu et al., 2023).

Methods	Retrain-free	Internal Encode	Task-adaptable	Enhanced Filtering
ERNIE	✗	✓	✓	✗
KEPLER	✗	✓	✓	✗
E-BERT	✓	✗	✗	✗
PELT	✓	✓	✗	✗
MapTuning	✓	✗	✓	✗
KEFF(ours)	✓	✓	✓	✓

Table 1: Comparison of KEFF with existing knowledge enhancement methods.

Existing knowledge-enhanced approaches typically train the PLM from scratch or fine-tune the PLM (Hu et al., 2023). For the paradigm of training the PLM from scratch, existing approaches typically introduce an external knowledge base, add knowledge-related objective tasks or modify knowledge-related loss functions during the training phase of the PLM, and subsequently train the entire PLM parameters (Zhang et al., 2019; Wang et al., 2021; Yamada et al., 2020). Fine-tuning PLMs usually update some of the PLM's parameters by training with the introduction of external knowledge (Kang et al., 2022). Both the above paradigms are effective in enhancing the capabilities of PLMs, but expend extensive computational cost and time for training.

Fortunately, some research explores knowledge enhancement without retraining or fine-tuning the PLM. (Lin et al., 2019a). In this scenario, existing methods freeze the PLM parameters, introduce additional networks for entity alignment training (Poerner et al., 2019; Ye et al., 2022; Zhang et al., 2023;

*Corresponding author

Findings of the Association for Computational Linguistics:
NAACL 2025, pages 3860–3871

Wang et al., 2020), and hence perform knowledge enhancement PLM. This paradigm significantly reduces the computational cost as well as the training time of knowledge-enhanced PLMs and has become a major research issue in recent years. However, despite the improved capabilities of PLM in this paradigm, existing methods still suffer from the following problems:

(1) Heavy reliance on the embedding information of entities in external knowledge bases. Existing methods first align the entity embedding space in the knowledge base with the PLM embedding space before introducing entity knowledge representation. If new entities are added to the knowledge base, they must be recoded, which reduces the efficiency of existing methods.

(2) Considering only knowledge enhancement without filtering. Existing approaches introduce knowledge representations of entities aligned with knowledge bases based on the original PLM inputs, but ignore a critical issue: the introduced knowledge representations of entities may become noise and disturb the original reasoning ability of the pretrained language model. Therefore, it is necessary to filter the introduced knowledge, i.e., selectively implement knowledge enhancement.

To address the above problems, we propose a novel knowledge enhancement and filtering framework that simultaneously consists of two modules: knowledge enhancement and knowledge enhancement filter. For knowledge enhancement, we present an **interesting and effective insight: there are redundant bits with small absolute values in the embedding space of PLM.** Modifying the values of these redundant bits has little or no negative impact on the performance of the PLM in downstream tasks. Building upon this crucial insight, we propose a knowledge enhancement method based on the redundant bits in the PLM embedding space. For knowledge-enhanced filtering, we combine a knowledge enhancement network with mask training, utilizing a single-layer classifier network as a filter. We summarize existing knowledge enhancement methods in Table 1 and provide a multifaceted comparison between our proposed KEFF and these methods.

Our main contributions are as follows:

- We present an effective and interesting insight: there are redundant bits with small absolute values in the embedding space of PLM. Exploring knowledge enhancement for these redundant bits appears to be a valuable research direction.
- We propose KEFF, the first framework to simultaneously consider both knowledge enhancement and knowledge enhancement filtering. KEFF includes a knowledge enhancement filter that allows for selective knowledge enhancement when pre-trained language models are applied to downstream tasks. Moreover, KEFF is a plug-and-play framework so it does not require retraining or fine-tuning of the PLM, thereby reducing computational overhead.
- We conduct experiments on four konwledge-driven downstream tasks. Compared with the state-of-the-art (SOTA) methods, Our method performs better in almost all downstream tasks and can achieve 1.7-2.5x computational cost reduction.

2 Related Work

Research on knowledge enhancement methods for PLM is mainly centered around models such as BERT (Devlin et al., 2018), and we categorize existing methods into three paradigms.

2.1 Train PLM from scratch

These methods generally modify the original training objectives of the PLMs and add additional loss function terms for knowledge enhancement (Hu et al., 2023). ERNIE (Zhang et al., 2019) utilizes large-scale corpora and knowledge graphs to train knowledge-enhanced PLMs. KEPLER (Wang et al., 2021) encodes entity descriptions as entity embeddings, and introduces a loss term for knowledge encoding (KE) in addition to the balance of training masked language models (MLMs). LUKE (Yamada et al., 2020) introduces an entity-aware self-attention mechanism based on an external knowledge base. Although such methods perform better on entity-related tasks, they require updating all PLM parameters, resulting in significant computational resource consumption and extended training time.

2.2 Fine-tuning the PLM

This paradigm of knowledge enhancement approach has less training overhead than training the entire PLM from scratch, but is mostly applied to specific downstream tasks (QA-GNN (Yasunaga

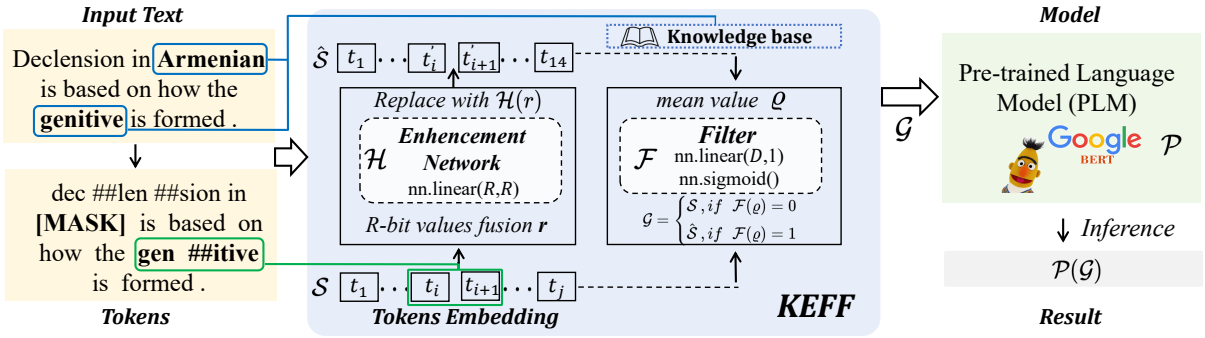


Figure 1: Overview of KEFF. KEFF is a plug-and-play framework and works without modifying PLM. For the token embedding of entities in the input text that are aligned with the knowledge base, Knowledge Enhancement is applied for mapping and transformation. Both the original input and the enhanced input are fed into the Filter separately, then final result is output. D represents the token embedding length of the PLM.

et al., 2021), CokeBERT (Su et al., 2021), etc.) Adam Roberts et al.(Roberts et al., 2020) employed the T5 model, fine-tuned specifically for answering open-domain questions. In contrast, CokeBERT(Su et al., 2021) incorporated special markers during the fine-tuning process to improve their performance on various downstream tasks.

2.3 Enhancing Frozen PLM with Networks

This paradigmatic approach to knowledge enhancement freezes the original PLMs and trains only the neural networks used for knowledge embedding. KagNet (Lin et al., 2019b) builds a structural common-sense knowledge graph outside of the BERT (Devlin et al., 2018) model to perform interpretable reasoning. K-Adapter (Wang et al., 2020) builds adapters for different tasks and uses different datasets to train the adapters separately for PLM reasoning on downstream tasks. E-BERT (Poerner et al., 2019) aligns entity encoding from external knowledge bases with PLM embedding, and introduces additional entity referents in addition to the original text during model reasoning. PELT (Ye et al., 2022) and MapTuning (Zhang et al., 2023) also adopt this additional introduction of entity referents. However, PELT (Ye et al., 2022) cannot be adapted for downstream tasks, while E-BERT (Poerner et al., 2019) and MapTuning (Zhang et al., 2023) need to rely on external knowledge base encoding. In contrast, our proposed method utilizes the PLM's own embeddings for knowledge enhancement. Moreover, existing approaches to knowledge enhancement can sometimes lead the original PLM to produce incorrect reasoning.

3 Methodology

In this section, we present KEFF, which contains two main modules: knowledge enhancement and knowledge enhancement filter. We illustrate the workflow of KEFF in Figure 1.

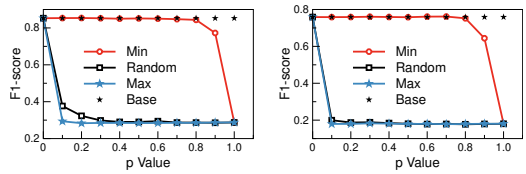
3.1 Preliminaries

Given an input sentence $\mathcal{S} = \{t_1, t_2, ..., t_N\} \in \mathbb{R}^{N \times W}$, where N is the number of tokens in the input sentence, W is the embedding dimension size of token in PLM. We define an external knowledge base as $\mathcal{B}$, The downstream task is defined as $\mathcal{D}$ and the PLM is defined as $\mathcal{P}$. For a given $\mathcal{S}$, assume that $\mathcal{S}$ contains j entities in $\mathcal{B}$, we define entity mention span $e_j = t_i^j, ..., t_{i+l}^j \subset \mathcal{S}$, where l is e_j's variable length.

3.2 Insight: Exploring Redundant Bits in PLM's Embedding Space

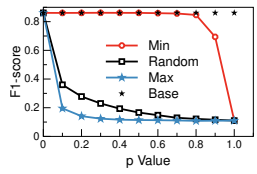
Existing approaches favor doing alignment from the knowledge coding space to the PLM coding space to achieve knowledge injection, and we note that there is a lack of research on the coding space of the PLM itself. Inspired by the existence of redundant neurons (activation value is very small) (Han et al., 2015; He et al., 2019) in neural networks, we consider the question: **Is there a certain number of redundant bits in the PLM's embedding space that can be used for knowledge enhancement ?**

We use BERT (Devlin et al., 2018) as the pre-trained language model $\mathcal{P}$, for given input sentence $\mathcal{S}$, we first find all the entity mentioned span $(e_1, e_2, ..., e_j)$, subsequently, given the ratio p, for all the tokens in e_j , we explore the locations in these tokens that have the smallest absolute value of

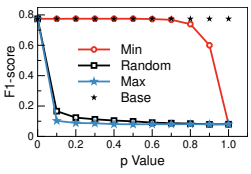


(a) FewRel 5-1

(b) FewRel 10-1



(c) Wiki80



(d) Wiki-ET

Figure 2: Insight illustration. (a) and (b) are implemented using FewRel 1.0, where N-K indicates the N-way K-shot setting. (c) and (d) are implemented using Wiki80 and Wiki-ET, respectively.

scale p and set the value of these locations to 0, then we apply the models to different downstream tasks $\mathcal{D}$. Figure 2 compares the performance of the original and processed BERT models on three entity-related downstream tasks: FewRel 1.0, Wiki80, and WikiET. Details of these tasks are provided in Section 4.

As shown in Figure 2, "Min" refers to setting the smallest bit of the token proportion for entity mentions to 0. When p increases from 0.1 to 1, the BERT model's F1-score gradually declines across four downstream tasks. However, within a certain smaller range, increasing p causes minimal performance drop and even some improvement. This reveals a key insight: **redundant bits with small absolute values** exist in the PLM's embedding space, and altering these values has little impact on the model's performance.

To further validate this, we tested two other settings: 1) Max, setting the largest bit of the token proportion to 0, and 2) Random, setting a random bit to 0. In both cases, the BERT model's performance significantly declined, even with small p values. For more insight experimental results, see the appendix for details.

3.3 Knowledge Enhancement

Based on our proposed insight, we give the knowledge enhancement method in KEFF. For a given pre-trained language model $\mathcal{P}$ with token embedding dimension size W and corresponding number of redundant bits R, where $R < W$, we perform knowledge enhancement on the values of the redundant bits using a simple affine transformation $\mathcal{H}$, with the size of the $\mathcal{H}$ network being $R \times R + R$. Similar to MapTuning, we introduce a variant of Masked Language Modelling (MLM) (Devlin et al., 2018), i.e., mentioning Masked Language Modelling (MMLM), where we freeze the PLM and train only $\mathcal{H}$ while masking only entity mentions in the text during training.

During training, for MMLM, we retain the original loss function, defined as $\mathcal{L}_{mmlm}$. For unmasked entity mention $e = \{t_1, t_2, ..., t_j\}$, we define $\mathcal{R}(t_i)$ as the value of redundant bits in the $i-$th token, and compute

$$r = \frac{1}{j}\sum_{i=1}^{j}\mathcal{R}(t_i) \tag{1}$$

The r represents the average of the redundant bit values of all the tokens in the entity mentioned span e, which we subsequently map using the network $\mathcal{H}$. After mapping, we use the mapped value $\mathcal{H}(r)$ to replace the redundant bit value of each token in e. In order to enable knowledge enhancement of the replaced embedding value $\mathcal{H}(r)$, while minimizing the redundant bits of the original tokens, we use KL scatter to constrain the distribution of $\mathcal{H}(r)$, while MSE is used to constrain the scale of $\mathcal{H}(r)$. We define the two loss functions $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$ separately:

$$\begin{aligned}\mathcal{L}_{KL} &= \sum_{i} r log(\frac{r}{\mathcal{H}(r)}) \\ \mathcal{L}_{MSE} &= \frac{1}{n}\sum_{i=1}^{n}(r_i - \mathcal{H}(r_i))^2\end{aligned} \tag{2}$$

To sum up, we present the final form of the loss function as follows:

$$\mathcal{L}_{KE} = \mathcal{L}_{MMLM} + \mathcal{L}_{KL} + \mathcal{L}_{MSE} \tag{3}$$

After $\mathcal{H}$ is trained, it can be applied to various downstream tasks $\mathcal{D}$ within $\mathcal{P}$. In addition to this, $\mathcal{H}$ can be fine-tuned using some of the downstream task $\mathcal{D}$'s training data when applied to the downstream task, and can therefore be adapted for different downstream tasks, which is the same as our description in Table 1.

To determine the optimal redundant bits R, we propose a approach named **Neuron Clipping**. We start by defining the redundant bit space as half the size of the PLM embedding space and train the knowledge enhancement network H within KEFF. After training, we iteratively prune the input to H

by zeroing the largest absolute values and removing the most affected outputs. The best-performing redundant bits during this process are selected as the optimal ones. We present the corresponding results in the appendix.

3.4 Knowledge Enhancement Filter

Based on our knowledge enhancement approach, we define the general form of the knowledge enhancement filter $\mathcal{F}$: $\mathcal{F}(*) = \{0, 1\}$, and we define that when $\mathcal{F}$ outputs 0, $\mathcal{P}$ reasons using the original $\mathcal{S}$. When $\mathcal{F}$ outputs 1, $\mathcal{P}$ reasons using the input $\hat{\mathcal{S}}$ enhanced with the knowledge enhancement by $\mathcal{H}$. However, there is a huge challenge with such an approach: we do not know the magnitude of the difference between $\mathcal{P}$ using the original input $\mathcal{S}$ and the knowledge-enhanced input $\hat{\mathcal{S}}$ until $\mathcal{P}$ is applied to a downstream task $\mathcal{D}$, and thus cannot tell in advance whether to perform knowledge enhancement or not.

To address this challenge, we propose a training method for knowledge-enhanced filters that incorporates references to MMLM. When $\mathcal{H}$ is trained, based on the data $\mathcal{B}_{sub} \subset \mathcal{B}$ in the partial external knowledge base $\mathcal{B}$, using only the loss function $\mathcal{L}_{mmlm}$ of the MMLM. We define $\mathcal{L}_{mmlm}(\mathcal{S})$ as the loss under the use of the original input S, and $\mathcal{L}_{mmlm}(\hat{\mathcal{S}})$ as the loss under the use of the knowledge-enhanced input $\hat{\mathcal{S}}$, and for each data in $\mathcal{B}_{sub}$ for each piece of data define the label L to be

$$L = \begin{cases} 1, & if\ \mathcal{L}_{MMLM}(\hat{\mathcal{S}}) \leq \mathcal{L}_{MMLM}(\mathcal{S}) \\ 0, & others \end{cases} \tag{4}$$

For the input of $\mathcal{F}$, we provide the following definition:

$$\varrho = \frac{1}{j} \sum_{i=1}^{j} Encode(t_i) \circ \mathcal{H}(r) \tag{5}$$

where $\circ$ denotes replacing the values of redundant bits in the embedding space of t_i with the mapped values $\mathcal{H}(r)$ obtained from the knowledge-enhanced network. Subsequently, we define the binary classification loss function $\mathcal{L}_{KEF}$ for $\mathcal{F}$:

$$\mathcal{L}_{KEF} = -\frac{1}{N} \sum_{i=1}^{N} [L \log(\mathcal{F}(\varrho)) + (1 - L) \log(1 - \mathcal{F}(\varrho))] \tag{6}$$

We define $\mathcal{F}$ as a single-layer fully connected network with an input dimension of R and an output dimension of 1. Upon completion of training, $\mathcal{F}$ is utilized in conjunction with $\mathcal{H}$ for the downstream task. We establish a threshold θ, where the final output is determined to be 1 if $\mathcal{F}(\varrho)$ meets or exceeds this threshold, and 0 otherwise. The procedures for KEFF's knowledge enhancement and filtering algorithms are delineated in Algorithm 1.

4 Experimental Setup

In our BERT$_{base}$ experiments, we set 149 redundant bits (out of 768 dimensions) and a filter threshold θ of 0.50. We provide more detailed experiment results in the appendix, such as optimal redundant position exploration and more details about our knowledge enhancement filter. Our code and datasets are available at https://github.com/tize-72/Keff.

4.1 Baselines

In this work, we focus on this paradigm of freezing PLM and using additional networks for knowledge enhancement, so we choose four SOTA approaches in this paradigm as baseline to compare with our proposed KEFF.

(1) E-BERT (Poerner et al., 2019): Uses a network to align entity vectors with BERT's word vectors, adding the aligned entity embeddings to the original input. (2) K-Adapter (Wang et al., 2020): Employs different adapter networks for various tasks to enhance knowledge. (3) PELT (Ye et al., 2022): Aggregates entity outputs from different contexts to construct entity embeddings, which are then added to the original input. (4) MapTuning (Zhang et al., 2023): Aligns entity vectors with BERT's space using an additional network, and incorporates MMLM during training, with the mapping network fine-tuned for downstream tasks.

4.2 Datasets and Metrics

Datasets. To validate the effectiveness of our proposed method, we selected four different entity-related datasets for our experiments. FewRel 1.0 (Han et al., 2018) is a large-scale supervised dataset for the task of few-shot relationship classification, covering 100 relationships with 700 instances of each. FewRel 2.0 (Gao et al., 2019) additionally introduces cross-domain and None-of-

the-above two challenges[1]. Wiki80 (Han et al., 2019) for entity-relationship categorization on full-volume data. For Entity typing, we experimented with Wiki-ET (Xin et al., 2022), which contains 68 entity types from Freebase. The scale of the datasets used and a summary of the tasks are provided in Table 2.

Dataset	Task	Size
FewRel 1.0	Few-Shot Relation Classification	70,000
FewRel 2.0	Few-Shot Relation Classification	70,000
Wiki80	Relation Classification	56,000
Wiki-ET	Entity Typing	68,242

Table 2: Summary of Dataset Scale and Tasks

Metrics. We use the same evaluation metrics as the existing work (Zhang et al., 2023), employing F1-score for relation classification tasks and accuracy for entity typing tasks. It is additionally noted that since both FewRel1.0 and FewRel2.0 are few-shot relation classification datasets, we employed the N-way K-shot method for testing.

4.3 PLM and Knowledge Base

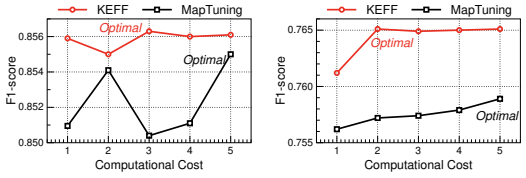
We use the BERT$_{base}$ (bert-base-uncased) (Devlin et al., 2018) model as our base PLM, using parameter files from publicly available data on Hugging Face. We use Wiki20M (Gao et al., 2021) as an external knowledge base. Since both MapTuning and E-BERT require external knowledge base encoding, we use TransE (Bordes et al., 2013) as a uniform standard.

5 Results

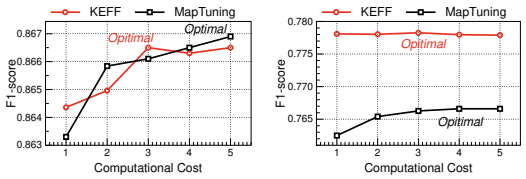
5.1 Main results

We report the experimental results in Table 3 and Table 4. For MapTuning(Zhang et al., 2023), we directly apply its mapping network to downstream tasks. To ensure a fair comparison, KEFF's knowledge enhancement network is also applied without additional fine-tuning.

Strong generalization capability for downstream tasks. We report the entity relationship classification results of KEFF and other baseline methods in the few-shot learning setting. Table 3 illustrates that KEFF achieves the best results in nearly all few-shot relationship classification tasks. Notably, while all baseline methods struggle with the FewRel 2.0 dataset, KEFF achieves a significant improvement, with an increase of up to 0.75 in this task. Table 4 illustrates the results for entity relationship classification under the full-data setting and entity typing. Although KEFF doesn't achieve the best results in the full-data setting, its performance is nearly on par with SOTA methods. For entity typing, most existing methods underperform, with K-Adapter showing a slight improvement over BERT$_{base}$. However, KEFF demonstrates a significant performance boost, achieving up to a 0.40 improvement.



(a) FewRel 5-1 (b) FewRel 10-1



(c) Wiki80 (d) Wiki-ET

Figure 3: Efficiency illustration. (a) and (b) are implemented using FewRel 1.0, where N-K indicates the N-way K-shot configuration. (c) and (d) are implemented using Wiki80 and Wiki-ET.

5.2 Efficiency Analysis

Lower costs with improved performance. To demonstrate the efficiency of our method, we compare it with MapTuning (Zhang et al., 2023), the best-performing baseline. We use the 5way1shot and 10way1shot settings of FewRel 1.0, along with the Wiki80 and Wiki-ET datasets for validation. Both methods employ an additional network for knowledge-enhanced mapping, but our affine transformation has significantly fewer parameters (149×149+149) compared to MapTuning's 768×128+768 network. We give the results in Figure 3, where "Optimal" denotes the point where the F1-score achieves the optimum. We can extract some important conclusions from Figure 3: (1) KEFF can achieve **better results with the same computational cost.** (2) KEFF achieves opti-

[1]Note that FewRel 1.0 and FewRel 2.0 do not make public the test sets they actually use, so all of our experiments are conducted with the validation set and some of the test set data that is currently publicly available.

Method	FewRel 1.0				FewRel 2.0			
	5-1	5-5	10-1	10-5	5-1	5-5	10-1	10-5
BERT_{base}	85.28	90.02	75.89	81.45	77.52	85.31	68.91	76.01
E-BERT	85.30	90.31	74.29	81.13	71.92	84.47	64.1	75.32
K-Adapter	81.5	89.81	70.59	79.43	69.52	83.97	63.6	74.52
PELT	83.4	90.11	73.09	80.33	71.42	84.27	62.9	74.52
MapTuning	85.5	90.91	75.89	**81.73**	73.52	85.07	65.7	75.92
KEFF	**85.63**	**91.15**	**76.51**	81.66	**77.91**	**86.03**	**69.08**	**76.76**
-w/o KE	85.28	90.02	75.89	81.45	77.52	85.31	68.91	76.01
-w/o KEF	85.54	91.10	76.32	81.45	77.79	85.91	69.01	76.57

Table 3: Comprehensive comparison results of our approach with the baseline approachs on FewRel 1.0 and FewRel 2.0. BERT_{base} serves as the base model and has been fine-tuned for different downstream tasks, E-BERT, K-Adapter, PELT, MapTuning, and our approach KEFF are based on BERT_{base} for knowledge enhancement. N-K indicates the N-way K-shot configuration. We boldface the best result for each method. KE and KEF represent the Knowledge Enhancement and Knowledge Enhancement Filter of KEFF respectively.

Method	Wiki80	Wiki-ET
BERT_{base}	86.12	77.43
E-BERT	85.39	77.06
K-Adapter	85.49	77.46
PELT	84.99	75.86
MapTuning	**86.69**	76.66
KEFF	86.67	**77.83**
-w/o KE	86.12	77.43
-w/o KEF	86.54	77.76

Table 4: Comprehensive comparison results of KEFF with the baseline approachs on Wiki80 and Wiki-ET.

Method	Wiki80	Wiki-ET
BERT_{base}	86.12	77.43
MapTuning	86.69	76.66
KEFF(ours)	86.67	77.83
MapTuning (FT)	87.99	78.73
KEFF (FT))	**88.01**	**79.92**

Table 5: Comprehensive comparison results of KEFF with MapTuning on Wiki80 and Wiki-ET after downstream tasks fine-tuning(FT).

mal results with about **1.7-2.5x less computational cost** compared to MapTuning.

5.3 Adaptability Analysis

Superior performance adaptation in downstream tasks. Existing approaches to knowledge enhancement using add-on networks under frozen PLM are rarely adapted to specific downstream tasks. E-BERT trains entity alignment before the model is used, and PELT directly constructs entity lookup tables. None of them can be adapted for downstream tasks. MapTuning utilizes mapping networks for knowledge enhancement training, and thus can be fine-tuned for downstream tasks. Our proposed knowledge enhancement method also has the ability to be fine-tuned to adapt to various downstream tasks. We use Wiki80 and Wiki-ET as test datasets and reporte the results of KEFF and MapTuning after fine-tuning to adapt to downstream tasks. Table 5 illustrates that KEFF effectively adapts to fine-tuning for downstream tasks and outperforms MapTuning, achieving a performance increase of up to 2.49.

5.4 Compatibility Analysis

Compatible with existing plug-and-play methods, achieving better results. Table 7 illustrates that KEFF can be perfectly combined with existing plug-and-play methods MapTuning to achieve better performance on downstream tasks. On the Wiki80 and WikiET, we achieved an improvement of up to 0.92. It should be noted that KEFF does not conflict with existing methods. We give certain reasons for this analysis: (1) KEFF is also a plug-and-play method, and can use the same knowledge basesuch as WiKi20M (Gao et al., 2021) as other similar methods. (2) Unlike existing methods, KEFF enhances knowledge directly within the embedding space of the PLM without adding extra entity tokens to the input text. This unique approach allows KEFF to be compatible with other existing plug-and-play methods.

5.5 Ablation Analysis

KE is the most important module in KEFF. We report the results of ablation experiments in Tables

Input	Groud truth	method	Predicted label	Logits	Filter output	Final result
Gorillaz released a single with James Murphy and André 3000 commissioned by Converse , titled " DoYaThings " on 23 February 2012.	**performer**	Bert_{base}	composer, performer, screenwriter has part, record label	7.387, 7.351, 2.764 2.303, 1.811	1	**perfomer**
		KEFF	**performer**, composer, screenwriter has part; record label	7.566, 7.176, 2.729 2.403, 1.891		
He married Caroline Macmillan , a daughter of Harold Macmillan and Lady Dorothy Cavendish, a daughter of the 9th Duke of Devonshire.	**child**	Bert_{base}	**child**, mother, spouse sibling, father	7.959, 7.938, 4.124 2.979, 2.605	0	**child**
		KEFF	mother, child, spouse sibling, father	7.967, 7.924, 4.127 2.976, 2.593		

Table 6: A case study of the KEFF on the Wiki80 dataset. Underlines indicate entities in the input. Groud truth denotes the true labeling of entity relationships in the text, and Filter output denotes the output of the Knowledge Enhancement Filter in KEFF, where an output of 1 means that the input is enhanced by the Knowledge Enhancement, and an output of 0 means that this KE's output is filtered and still employs the original input.

Method	Wiki80	Wiki-ET
BERT_{base}	86.12	77.43
MapTuning	86.69	76.66
KEFF(ours)	86.67	77.83
-w/ MapTuning	**87.04**	**78.26**

Table 7: Effectiveness of KEFF in combination with existing methods on Wiki80 and Wiki-ET.

Method	Wiki80(F1)
KEFF (ours)	**88.01**
Llama3.1-8B	42.43
Llama2-7B	10.49
Qwen2.5-32B	63.77
Qwen2-7B	33.63
Phi3-14B	31.93
Mistral-7B	25.30
Qwen2.5-72B	66.48

Table 8: Performance comparison of KEFF with different LLMs on the Wiki80 dataset.

3 and 4. After removing Knowledge Enhancement (KE), KEFF's performance decreased significantly across all downstream tasks, particularly in the 5-way 5-shot scenario of FewRel 1.0 dataset, where the performance drop reached 1.13. This indicates that the KE plays a substantial role in promoting KEFF's overall performance.

The introduction of the Filter makes knowledge enhancement selective. The results show that after removing the Knowledge Enhancement Filter(KEF), KEFF's performance also declined, but the extent of the decrease was relatively small. IThis indicates that the Filter brought selectivity to KE's knowledge enhancement, thereby improving KEFF's final performance.

5.6 LLMs Comparison Analysis

KEFF outperforms larger parameter-scale large language models(LLMs). To explore the performance differences between KEFF and larger parameter-scale LLMs, we tested seven LLMs on the Wiki80 dataset, with results shown in Table 8. Among all the models, Qwen2.5-72B achieved the highest F1-score of 66.48, still far behind KEFF's 88.01. This suggests that LLMs cannot achieve optimal performance on related entity classification tasks without fine-tuning. However, the computational cost of fine-tuning these models is substantial, presenting a significant trade-off.

5.7 Case Study

The test example of KEFF applied to the downstream task Wiki80 is shown in Table 6. The filter output in Table 6 represents the output of the Knowledge Enhancement Filter (KEF), where an output of 1 indicates that the results from the KE plugin are adopted, and if the output is 0, the output from Bert_{base} is utilized. In the first example, the first predicted label "performer" of our KEFF can be achieved through the Knowledge Enhancement Filter, so we adopt the KE's result as the final output. In the second example, we first observe that the difference between the predicted logits is very small, indicating that the KEFF plug-and-play strategy does not significantly deviate the predictions of the base model.

Although in very few cases KE may yield incorrect results, such as "mother", the filter discards the KEFF strategy and opting for the original Bert_{base}'s output to ensure the accuracy of the final result. This demonstrates that the plug-and-play KEFF improves the knowledge enhancement results of the base model.

6 Conclusion

In this paper, we present a key insight: the PLM's embedding space contains redundant bits with small absolute values. Based on this, we propose the first plug-and-play framework that integrates knowledge enhancement and knowledge enhancement filter, named **KEFF**. Extended experiments on multiple downstream tasks indicate that: 1) KEFF outperforms the current state-of-the-art methods in terms of performance, 2) simultaneously achieves a 1.7 to 2.5x reduction in computational cost, 3) adapts to various entity-related downstream tasks, and 4) can be well integrated with other existing plug-and-play knowledge enhancement methods.

Limitations

This work focuses on improving the performance of PLMs on entity-related downstream tasks. Due to the current prevalence of decoder-only architectures in LLMs, this work cannot be directly transferred to these large model architectures at present. However, the insights we propose may provide some assistance for knowledge augmentation at the encoding level in LLMs.

Acknowledgments

The research is partially supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang, 2023C01029, National Key R&D Program of China under Grant No. 2021ZD0302900 and No. 2021ZD0110400, China National Natural Science Foundation with No. 62132018, The Plans for Major Provincial Science&Technology Projects (No.: 202303a07020006) and Fundamental Research Funds for the Central Universities.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Le Centre pour la Communication Scientifique Directe - HAL - Inria,Le Centre pour la Communication Scientifique Directe - HAL - Inria*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. *arXiv preprint arXiv:1807.04905*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

Tianyu Gao, Xu Han, Keyue Qiu, Yuzhuo Bai, Zhiyu Xie, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Manual evaluation matters: reviewing test protocols of distantly supervised relation extraction. *arXiv preprint arXiv:2105.09543*.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel 2.0: Towards more challenging few-shot relation classification. *arXiv preprint arXiv:1910.07124*.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, and Maosong Sun. 2019. Opennre: An open and extensible toolkit for neural relation extraction. *arXiv preprint arXiv:1909.13078*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. 2019. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4340–4349.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*.

Minki Kang, Jinheon Baek, and Sung Ju Hwang. 2022. Kala: knowledge-augmented language model adaptation. *arXiv preprint arXiv:2204.10555*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10965–10973.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019a. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019b. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. E-bert: Efficient-yet-effective entity embeddings for bert. *arXiv preprint arXiv:1911.03681*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

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.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2022. Improving neural fine-grained entity typing with knowledge attention. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Jian Yang, Gang Xiao, Yulong Shen, Wei Jiang, Xinyu Hu, Ying Zhang, and Jinghui Peng. 2021. A survey of knowledge enhanced pre-trained models. *arXiv preprint arXiv:2110.00269*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*.

Deming Ye, Yankai Lin, Peng Li, Maosong Sun, and Zhiyuan Liu. 2022. A simple but effective pluggable entity lookup table for pre-trained language models. *arXiv preprint arXiv:2202.13392*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Huadong Wang, Deming Ye, Chaojun Xiao, Xu Han, Zhiyuan Liu, Peng Li, Maosong Sun, et al. 2023. Plug-and-play knowledge injection for pre-trained language models. *arXiv preprint arXiv:2305.17691*.

A Knowledge Enhancement and Filtering

Algorithm 1 Knowledge enhancement and filtering

Input : Knowledge enhancement network H, a token sequence S of length n, an entity mention token list E of length l, a filter F, threshold θ

Function KEFF-enhance(S, E, H)**:**

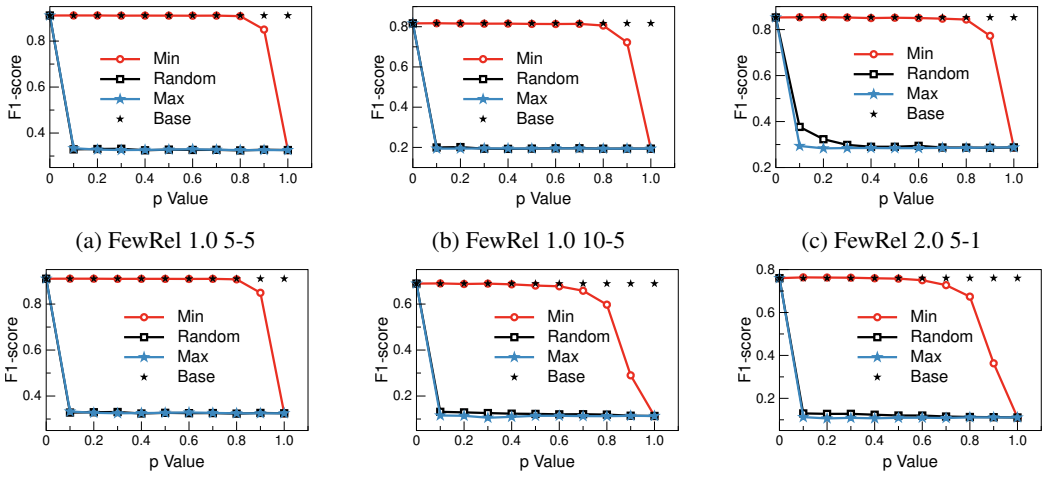
$V \leftarrow []$, $D \leftarrow []$;
for $i \leftarrow 1$ **to** n **do**
  if S_i ***in*** E **then**
    $d \leftarrow$ indices of minimum values in S_i;
    $v \leftarrow$ minimum r-bit absolute value of S_i;
    V.append(v), D.append(d);
$r \leftarrow H\left(\text{Average}\left(\sum_{j=1} V[j]\right)\right)$, $d \leftarrow$ length of D;
for $i \leftarrow 1$ **to** n **do**
  for $z \leftarrow 1$ **to** d **do**
    if S_i ***in*** E **then**
      replace($S_i[D_z]$, r);

Function KEFF-filter(S, E, D, F, θ)**:**

$r \leftarrow$ calculate by equation(5), $d \leftarrow$ length of D;
for $i \leftarrow 1$ **to** n **do**
  for $z \leftarrow 1$ **to** d **do**
    if S_i ***in*** E **then**
      if $F(r) \geq \theta$ **then**
        replace($S_i[D_z]$, r);

B Insight Experimental Results

We provide here more results of validation experiments for insight. We implement our experiments under two settings (5way1shot and 10way5shot) for FewRel 1.0. For the four settings of FewRel 2.0 (5way1shot, 5way5shot, 10way1shot, 10way5shot) we also give the results of validation experiments.



(a) FewRel 1.0 5-5 (b) FewRel 1.0 10-5 (c) FewRel 2.0 5-1

(d) FewRel 2.0 5-5 (e) FewRel 2.0 10-1 (f) FewRel 2.0 10-5

Figure 4: Insight illustration. (a) and (b) are implemented using FewRel 1.0, where N-K indicates the N-way K-shot setting. (c), (d), (e) and (f) are implemented using FewRel 2.0.

As shown in Figure 4, we set the minimum position of the absolute value of the proportion of tokens included in entity mentioned span that are p to 0. The F1-score of the PLM on the downstream task does not decrease significantly when p increases over a small range, and even improves at certain values of the p position. Similarly, to emphasize the effectiveness of our proposed insight, we set both Max and Random settings for comparison. It can be seen that with both settings, PLM's performance on the downstream task is still significantly negatively affected even when p is very small.

C Optimal Redundant Position Exploration

We provide a detailed description of the proposed Neuron Clipping method and present a comparative analysis of its efficiency.

C.1 Neuron Clipping

We give the experimental results of the exploration of the optimal redundant bits. In order to obtain the optimal redundant bits in the coding space of PLM more rapidly, we propose an innovative method based on neuronal clipping. We take the BERT model as an example, whose coding space is 768 bits in size. We first define the size of the redundant bit space of BERT as 384 bits (half of the original), and subsequently, we train the knowledge enhancement network in KEFF based on this redundant bit size with one-tenth of the external knowledge base dataset as the training set.

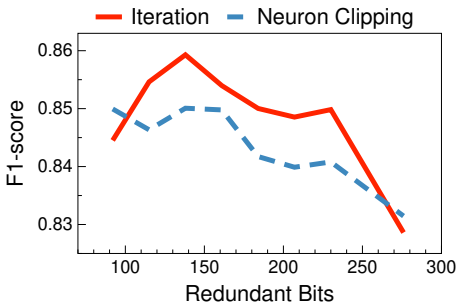


Figure 5: Exploring optimal redundant bits for iteration and neruon clipping.

After obtaining the knowledge enhancement network, we iterate the redundant bits in a certain step size. During the iteration of the redundant bits, we set the part of the original input of the knowledge enhancement network with a large absolute value to 0 and compare it with the result of the original input in the knowledge enhancement network. We **remove the highly affected part of the knowledge enhancement network** and use the remaining part as the output of the knowledge enhancement network in KEFF.

For comparison, we also use one-tenth of the external knowledge base dataset as a training set, and iterate over the redundant bits for a certain number of steps to train several different knowledge enhancement networks.

We use Wiki80 as a test set. Validating with these two different approaches, it can be observed from Figure 5 that both the neuron clipping based approach and the iterative approach achieved the

F1-score maximum at roughly the same location.

C.2 Efficiency

Figure 6 shows a comparison of the overhead of the two methods, and it can be seen that our proposed neuron clipping-based optimal redundancy bit exploration method achieves an overhead reduction of 8.0x for the data size, as well as a training overhead reduction of 3.6x.

It should be noted that the computitional cost spent on the optimal redundant bits exploration experiment is much smaller than the overall computitional cost of KEFF.

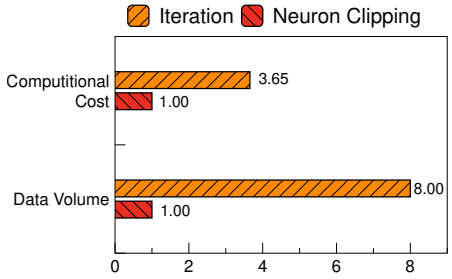


Figure 6: Comparison of training data overhead and computational cost for iteration with neruon clipping.

D Details about Filter

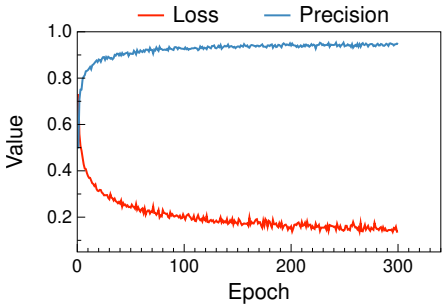


Figure 7: The precision and the training loss of the filter.

In this section, we present the precision of the filter along with the corresponding training loss. Additionally, we provide the specific training parameters and other relevant details of the training process for the filter.

Parameter	Value
Learning Rate	0.01
Batch Size	256
Epochs	300
Dataset	Wiki20M

Table 9: Training parameters of Filter

In Table 1, we provide the specific parameters used for training the Filter. It is important to note that we employed the Wiki20M dataset. However, we did not use the entire dataset due to its large size. Instead, we sampled 1% of the Wiki20M dataset for training the Filter, as the Filter's model structure is relatively simple.