DICT-BERT: ENHANCING LANGUAGE MODEL PRE-TRAINING WITH DICTIONARY

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-trained language models (PLMs) aim to learn universal language representations by conducting self-supervised training tasks on large-scale corpora. Since PLMs capture word semantics in different contexts, the quality of word representations highly depends on word frequency, which usually follows a heavy-tailed distributions in the pre-training corpus. Therefore, the embeddings of rare words on the tail are usually poorly optimized. In this work, we focus on enhancing language model pre-training by leveraging definitions of the rare words in dictionaries (e.g., Wiktionary). To incorporate a rare word definition as a part of input, we fetch its definition from the dictionary and append it to the end of the input text sequence. In addition to training with the masked language modeling objective, we propose two novel self-supervised pre-training tasks on word and sentence-level alignment between input text sequence and rare word definitions to enhance language modeling representation with dictionary. We evaluate the proposed Dict-BERT model on the language understanding benchmark GLUE and eight specialized domain benchmark datasets. Extensive experiments demonstrate that Dict-BERT can significantly improve the understanding of rare words and boost model performance on various NLP downstream tasks.

1 INTRODUCTION

In recent years, pre-training language models (PLMs) have revolutionized the field of natural language processing (NLP), yielding remarkable performance on various downstream NLP tasks (Qiu et al., 2020). For example, BERT (Devlin et al., 2019) and its novel variants such as RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019) capture syntactical and semantic knowledge mainly from the pre-training task of masked language modeling (MLM). However, these PLMs suffer from lacking domain-specific knowledge when completing many real-world tasks. To address this issue, many existing methods have incorporated domain knowledge from external resources to enrich the language representation, ranging from linguistic (Wang et al., 2021a), commonsense (Guan et al., 2020), factual (Wang et al., 2021b), to domain-specific knowledge (Liu et al., 2020; Yu et al., 2020a).

Nevertheless, rare words (Schick & Schütze, 2020) and unseen words (Cui et al., 2021) are still blind spots of pre-trained language models when fine-tuned on downstream NLP tasks. For instance, in a dialogue system, users often talk with chatbots about latest news and the recently hot topics, e.g., "Covid-19", which may not appear in the pre-training corpus. Since PLMs capture word semantics in different contexts, as a consequence, PLMs usually perform poorly when a user mentions such novel words (Wu et al., 2021; Cui et al., 2021; Ruzzetti et al., 2021). As indicated by Wu et al. (2021), the quality of word representations highly depends on the word frequency in the pre-training corpus, which typically follows a heavy-tail distribution. Therefore, a large proportion of words appear very few times and the embeddings of these rare words are poorly optimized (Gong et al., 2018; Schick & Schütze, 2020). Such embeddings usually carry inadequate semantic meaning, which complicate the understanding of input text, and even hurt the pre-training of the entire model.

In this work, we focus on enhancing language model pre-training by leveraging rare word definitions in English dictionaries (e.g., Wiktionary). We append the definitions of rare words to the end of the input text and encode the whole sequence with Transformer. The pre-training tasks are mainly based on the alignment between input text and the appended word definitions, some of which are randomly sampled polluted words and don't explain the input. We propose two types of pre-training objectives:

1) a word-level contrastive objective aims to maximize the mutual information between Transformer representations of a rare word appeared in the input text sequence and its dictionary definition. 2) a sentence-level discriminative objective aims at learning to differentiate between correct and polluted word definitions. During downstream fine-tuning, in order to avoid the appended rare word definitions diverting the sentence from its original meaning, we employ a knowledge attention mechanism that makes word definitions only visible to the corresponding words in the input text sequence. We name our method Dict-BERT. Notably, Dict-BERT is general and model-agnostic, in the sense that any pre-trained language model (e.g., BERT, RoBERTa) suffices and can be used.

Overall, our main contributions can be summarized as follows:

1. We are the first work to integrate word definitions in a dictionary into PLMs.
2. We propose two novel pre-training tasks on word and sentence-level alignment between input text sequence and rare word definitions to enhance language modeling with dictionary.
3. We evaluate Dict-BERT on the GLUE benchmark (Wang et al., 2019) and our model pre-trained from scratch can improve accuracy by +1.15% on average over BERT baseline.
4. We follow the domain adaptive pre-training (DAPT) setting (Gururangan et al., 2020), where language models are continuously pretrained with in-domain data. We evaluate Dict-BERT on eight specialized domain benchmark datasets. Our method can improve F1 score by +0.5% and +0.7% on average over BERT-DAPT and RoBERTa-DAPT baselines.

2 Related Work

Representation of rare words in language models. Pre-trained language models capture word semantics in different contexts to address the issue of polysemous and the context-dependent nature of words. Therefore, the quality of word representations highly depends on the word frequency in the corpus, which often follows a heavy-tail distribution. Many recent works have shown rare words that are not frequently covered in the corpus can hinder the understanding of specific yet important sentences Schick & Schütze (2020); Wu et al. (2021); Ruzzetti et al. (2021). Due to the poor quality of rare word representations, the pre-training model built on top of it suffers from noisy input semantic signals which lead to inefficient training. Gao et al. (2019) provides a theoretical understanding of the rare word problem, which illustrates that the problem lies in the sparse stochastic optimization of neural networks. Schick & Schütze (2020) adapt attentive mimicking to explicitly learn rare word embeddings to language models. Specifally, it introduces one-token approximation, a procedure that uses attentive mimicking even when the underlying language model uses subword-based tokenization. Wu et al. (2021) proposes to take notes for rare words on the fly (TNF) during pre-training. Specifically, TNF maintains a note dictionary and saves a rare word's contextual information as notes when the rare word occurs in a sentence. When the same rare word occurs again during training, the note information saved beforehand can be employed to enhance the semantics of the current sentence. Different from Wu et al. (2021) which maintains a fixed vocabulary of rare words during pre-training and fine-tuning, our method can dynamically adjust the vocabulary of rare words, obtain and represent their definitions in a dictionary in a plug-and-play manner.

Language model pre-training and knowledge-enhanced methods Recent years have seen substantial pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) have achieved remarkable performance in various NLP downstream tasks. However, these PLMs suffer from lacking domain-specific knowledge when completing many real-world tasks (Yu et al., 2020b). For example, BERT can not give full play to its value when dealing with electronic medical record analysis tasks in the medical field (Liu et al., 2020). A lot of efforts have been made on investigating how to integrate knowledge into PLMs (Yu et al., 2020a; Liu et al., 2021; Gunel et al., 2020; Xiong et al., 2020; Guan et al., 2020; Zhou et al., 2021). Overall, these approaches can be grouped into two categories: The first one is to explicitly inject entity representation into PLMs, where the representations are pre-computed from external sources (Zhang et al., 2019; Liu et al., 2021). For example, KG-BART encoded the graph structure of KGs with knowledge embedding algorithms like TransE (Bordes et al., 2013), and then took the informative entity embeddings as auxiliary input (Liu et al., 2021). However, the method of explicitly injecting entity representation into PLMs has been argued that the embedding vectors of words in text and entities in KG are obtained in separate ways, making their vector-space inconsistent (Liu et al., 2020). The second one is to implicitly model knowledge information into PLMs by performing knowledge-related tasks, such

as concept order recovering (Zhou et al., 2021), entity category prediction (Yu et al., 2020a). For example, CALM proposed a novel contrastive objective for packing more commonsense knowledge into the parameters, and jointly pre-trained both generative and contrastive objectives for enhancing commonsense language understanding and generation tasks (Zhou et al., 2021).

3 PROPOSED METHOD

In this section, we introduce the details of our model Dict-BERT. We first describe the notations and how to incorporate rare word definitions as a part of input. Then we detail the two novel self-supervised pre-training objectives. Finally, we introduce the knowledge attention during fine-tuning.

3.1 NOTATION AND PROBLEM DEFINITION

Given the input text sequence $X = [\text{CLS}, x_1, x_2, \cdots, x_L, \text{SEP}]$ with L tokens, a language model f_{LM} produces the contextual word representation $f_{LM}(X) = [h_{\text{CLS}}, h_1, h_2, \cdots, h_L, h_{\text{SEP}}]$. For a specific downstream task, a header function f_H further uses $f_{LM}(X)$ and generates the prediction as $f_H(h_{\text{CLS}})$ for sequence classification or $f_H([h_{\text{CLS}}, h_1, h_2, \cdots, h_L, h_{\text{SEP}}])$ for token classification.

The goal of our work is to learn better contextual word representation $f_{LM}(x)$ by leveraging definitions of the rare words in dictionaries (e.g., Wiktionary). Suppose $S = [s_1, \cdots, s_K]$ and $C = [c^{(1)}, \cdots, c^{(K)}]$ are the sets of rare words in the input text sequence X and their definitions in the dictionary. When a rare word s_i appears in the input text sequence, we fetch its definition from the dictionary as $c^{(i)} = [c_1^{(i)}, \cdots, c_{N_1}^{(i)}]$ with N_1 tokens, and append it to the end of the input text sequence. So, an input sequence X with appended definitions of K rare words S_{rare} can be written as: $\text{input} = [\text{CLS}, x_1, x_2, ..., x_L, \text{SEP}^{(1)}, c_1^{(1)}, c_2^{(1)}, ..., c_{N_1}^{(1)}; ...; \text{SEP}^{(K)}, c_1^{(K)}, c_2^{(K)}, ..., c_{N_K}^{(K)}, \text{SEP}]$, And the corresponding contextual representation generated from the language model f_{LM} as: $f_{LM}(\text{input}) = [h_{\text{CLS}}, h_1, h_2, \cdots, h_L, h_{\text{SEP}}^{(1)}, h_1^{(1)}, \cdots, h_{N_1}^{(1)}; \cdots\cdots; h_{\text{SEP}}^{(K)}, h_1^{(K)}, \cdots, h_{N_K}^{(K)}, h_{\text{SEP}}]$. For a specific downstream task, a header function f_H still uses $f_{LM}(X)$ to generate the prediction as $f_H(h_{\text{CLS}})$ for sequence classification or $f_H([h_{\text{CLS}}, h_1, h_2, \cdots, h_L, h_{\text{SEP}}^{(1)}])$ for token classification.

3.2 CHOOSING THE RARE WORDS

There are different ways to choose the rare word set S in a pre-training corpus. One way is to use a pre-defined absolute frequency value as the threshold. Wu et al. (2021) used 500 as the threshold to divide frequent words and rare words, and maintained a fixed vocabulary of rare words during pre-training and fine-tuning. However, rare words can vary greatly in different corpora. For example, rare words in the medical domain are very different from those in general domain (Lee et al., 2020). Besides, keeping a large threshold for a small downstream datasets makes the vocabulary of rare words too large. For example, only 51 words in the RTE dataset have a frequency of more than 500.

Therefore, we propose to choose specialized rare words for each pre-training corpus and downstream tasks. Specifically, we ranked all word frequency from smallest to largest, and add them to the list one by one until the word frequency of the added word reaches 10% of the total word frequency. Compared with Wu et al. (2021) which maintained a fixed vocabulary, our method can dynamically adjust the vocabulary of rare words, obtain and represent their definitions in dictionary in a plug-and-play manner. To fetch the definition of rare words, we leverage the largest online dictionary, i.e., Wiktionary, and collected a dump of Wiktionary[1] which includes definitions of 999,614 concepts.

3.3 PRELIMINARY: BERT PRE-TRAINING

We use the BERT (Devlin et al., 2019) model as an example to introduce the basics of the model architecture and training objective of PLMs. BERT is developed on a multi-layer bidirectional Transformer (Vaswani et al., 2017) encoder. The Transformer encoder is a stack of multiple identical layers, where each layer has two sub-layers: a self-attention sub-layer and a position-wise feed-forward sub-layer. The self-attention sub-layer produces outputs by calculating the scaled dot

[1] https://www.wiktionary.org/

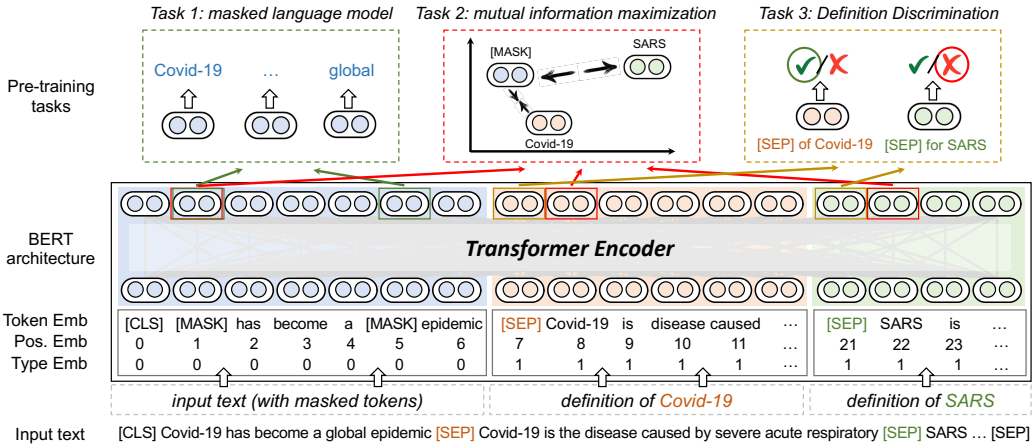


Figure 1: The overall architecture of Dict-BERT. The definitions of rare words are appended to the end of input text. In additional to training with masked language modeling, Dict-BERT performs two novel self-supervised learning tasks: word-level mutual information maximization (§3.4.1) and sentence-level definition discrimination (§3.4.2). "SARS" is a negatively sampled rare word.

products of queries and keys as the coefficients of the values, i.e.,

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V. \tag{1}$$

Q(Query), K(Key), V(Value) are the hidden representations produced by the previous self-attention layer and d is the dimension of the hidden representations. Transformer also extends the aforementioned self-attention layer to a multi-head self-attention layer version in order to jointly attend to information from different representation subspaces.

BERT uses the Transformer model as its backbone neural network architecture and trains the model parameters with the masked language modeling (MLM) objective on large text corpora. In the masked language modeling task, a random sample of the words in the input text sequence is selected. The selected positions will be either replaced by special token [MASK], replaced by randomly picked tokens or remain the same. The objective of masked language modeling is to predict words at the masked positions correctly given the masked sentences.

3.4 Dict-BERT: Language Model Pre-training with Dictionary

Dict-BERT is based on the BERT architecture, which can be initialized either randomly or from a pre-trained checkpoint with the same structure. It is worth noting that we slightly modified the type embedding, where the type embedding of the input text sequence is all 0, and the type embedding of the dictionary definitions is all 1. Besides, we used the absolute positional embedding.

We represent each input text sequence and dictionary definitions pair as a tuple (X, C). The semantics of a word in the input text depends on the current context, while the semantics of a word in the dictionary is standardized by linguistic experts. In order to better align the representations between them, we propose two novel pre-training tasks on word and sentence-level alignment between input text sequence and rare word definitions to enhance PLMs with dictionary.

3.4.1 Word-level Mutual Information Maximization.

Recently, there has been a revival of approaches inspired by the InfoMax principle (Oord et al., 2018; Tschannen et al., 2020): maximizing the mutual information (MI) between the input and its representation. MI measures the amount of information obtained about a random variable by observing another random variable. As the input text sequence and rare word definitions are obtained from different sources, in order to better align the representations, we proposed to maximize the MI between a rare word x_i in the input sequence and its well-defined meaning in the dictionary $c^{(i)}$, with

joint density $p(x_i, c^{(i)})$ and marginal densities $p(x_i)$ and $p(c^{(i)})$, is defined as the Kullback–Leibler (KL) divergence between the joint and the product of the marginals,

$$I(x_i; c^{(i)}) = D_{KL}(p(x_i, c^{(i)})||p(x_i)p(c^{(i)})) = \mathbb{E}_{p(x_i, c^{(i)})}[\log \frac{p(x_i, c^{(i)})}{p(x_i)p(c^{(i)})}]. \tag{2}$$

The intuition of maximizing mutual information between a rare word appeared in the input text sequence and its definitions in the dictionary is to encode the underlying shared information and align the semantic representation between the contextual meaning and well-defined meaning of a word. Nevertheless, estimating MI in high-dimensional spaces is a notoriously difficult task, and in practice one often maximizes a tractable lower bound on this quantity (Poole et al., 2019). Intuitively, if a classifier can accurately distinguish between samples drawn from the joint $p(x_i, c^{(i)})$ and those drawn from the product of marginals $p(x_i)p(c^{(i)})$, then x_i and $c^{(i)}$ have a high mutual information.

In order to approximate the mutual information, we adopted InfoNCE (Oord et al., 2018), which is one of the most commonly used estimators in the representation learning literature, defined as

$$I(x_i; c^{(i)}) \geq \mathbb{E}[\frac{1}{K}\sum_{i=1}^{K} \log \frac{e^{f_{\text{MI}}(h_i, h^{(i)})}}{\frac{1}{K}\sum_{j=1}^{K} \mathbb{1}_{[j \neq i]} e^{f_{\text{MI}}(h_i, h^{(j)})}}] \triangleq I_{NCE}(x_i; c^{(i)}), \tag{3}$$

where the expectation is over K independent samples $\{(h_i, h^{(i)})\}_{i=1}^{K}$ from the joint distribution $p(x_i, c^{(i)})$ (Poole et al., 2019). Intuitively, the critic function $f_{\text{MI}}(\cdot)$ measures the similarity (e.g., inner product) between two word representations. The model should assign high values to the correct pair $(h_i, h^{(i)})$, and low values to all other pairs. We compute InfoNCE using Monte Carlo estimation by averaging over multiple batches of samples (Chen et al., 2020). By maximizing the mutual information between the encoded representations, we extract the underlying latent variables that the rare words in the input text sequence and their dictionary definitions have in common.

3.4.2 Sentence-level Definition Discrimination

Instead of locally aligning the semantic representation, learning to differentiate between correct and polluted word definitions helps the language model capture global information of input text and dictionary definitions. We denote the set of definitions from rare words in the input text as C. We then sample a set of "polluted" definitions from dictionary by replacing C with probability 50% with a different word randomly sampled from the entire vocabulary together with its definition. Since the last layer representation on the special token [SEP] is the fused representation of a word definition, we apply a multi-layer perception (MLP) as a binary classifier $f_{\text{MLP}}(\cdot)$ to predict whether the appended definition is for a rare word ($y = 1$) or any polluted one ($y = 0$) in the input text sequence. Therefore, the discriminative objective can be formally defined as follows,

$$\mathcal{L}_{\text{DD}} = -\mathbb{E}\sum_{i=1}^{r_T} \log p(y | f_{\text{MLP}}(h_{\text{SEP}}^{(r_i)}). \tag{4}$$

3.4.3 Overall objective.

Now we present the overall training objective of Dict-BERT. To avoid catastrophic forgetting (McCloskey & Cohen, 1989) of general language understanding ability, we train the masked language modeling together with word-level mutual information maximization (MIM) and definition discrimination (DD) tasks. We denote $\mathcal{L}_{\text{MIM}}$ as the loss function of the MIM task which is the opposite of expectation in Eq.3. Hence, the overall learning objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \lambda_1 \mathcal{L}_{\text{MIM}} + \lambda_2 \mathcal{L}_{\text{DD}} \tag{5}$$

where λ_1, λ_2 are introduced as hyperparameters to control the importance of each task.

3.5 Dict-BERT Fine-tuning with Knowledge-visible Attention

Most existing works use the final hidden state of the first token (i.e., the [CLS] token) as the sequence representation (Devlin et al., 2019; Liu et al., 2019). For a sequence classification task, a multi-layer perception network function f_H takes the output of f_{LM} as input and generates the prediction as $f_H(h_{\text{CLS}})$. Notably, when fine-tuning a language model on downstream tasks, there could be many rare/unseen words in the dataset. Therefore, in the fine-tuning stage, when encountering a rare word in the input text, we append its definition to the end of input text, just like what we did in pre-training.

Table 1: Performance of different models on GLUE tasks. BERT § indicates the performance reported in Wu et al. (2021); The "dict in PT/FT" indicates whether to use the dictionary during pre-training/fine-tuning. Each configuration is run five times with different random seeds, and the average of these five results on the validation set is reported in the table. **We note that** our code is implemented on Huggingface Transformer (Wolf et al., 2020). The performance of our implemented BERT is consistent with the official performance, but it is slightly lower than the performance reported by Wu et al. (2021). Since no open-source code is released by BERT-TNF (Wu et al., 2021), we reported both their performance (BERT §) and our implemented performance. We computed the relative improvement (Δ) of BERT-TNF and Dict-BERT compared with the original BERT.

Methods	Dict in		MNLI	QNLI	QQP	SST	CoLA	MRPC	RTE	STS-B	Avg	Δ
	PT	FT	Acc.	Acc.	Acc.	Acc.	Matthews	Acc.	Acc.	Pearson		
BERT §	×	×	85.00	91.50	91.20	93.30	58.30	88.30	69.00	88.50	83.10	-
BERT-TNF §	√	√	85.00	91.00	91.20	93.20	59.50	89.30	73.20	88.50	83.90	+0.80
BERT (ours)	×	×	84.12	90.69	90.75	92.52	58.89	86.17	68.67	89.39	82.65	-
Dict-BERT-F	×	√	84.19	90.94	90.68	92.59	59.16	85.75	68.10	88.72	82.51	-0.14
Dict-BERT-P	√	×	84.33	91.02	90.69	92.62	60.44	86.81	**73.86**	**89.81**	83.70	+1.05
⊢ w/o MIM	√	×	84.24	90.79	90.24	92.22	60.14	87.03	73.79	89.67	83.52	+0.87
⊢ w/o DD	√	×	84.18	90.54	90.30	92.39	61.49	86.49	71.89	89.60	83.36	+0.71
Dict-BERT-PF	√	√	**84.34**	**91.20**	**90.81**	**92.65**	**61.68**	**87.21**	72.89	89.68	**83.80**	**+1.15**
⊢ w/o MIM	√	√	84.22	90.67	90.66	92.53	61.58	87.20	71.58	89.37	83.47	+0.82
⊢ w/o DD	√	√	84.16	90.21	90.78	92.39	61.14	87.19	71.84	89.24	83.37	+0.72

However, the appended dictionary definitions may change the meaning of the original sentence since the [CLS] token attend information from both input text and dictionary description. As pointed in Liu et al. (2020) and Xu et al. (2021), too much knowledge incorporation may divert the sentence from its original meaning by introducing a lot of noise. This is more likely to happen if there are multiple rare words in the input text. To address this issue, we adopt the visibility matrix (Liu et al., 2020) to limit the impact of definitions on the original text. In BERT, an attention mask matrix is added with the self-attention weights before $\mathrm{softmax}$. If token j is not supposed to be visible to token i, we add an -∞ value in attention matrix (i, j).

As shown in Figure 2, we modify the attention mask matrix such that a token i can attend to another token j only if: (1) both tokens belong to the input text sequence X, or (2) both tokens belong to the definition c of the same rare word, or (3) i is a rare word in the input text X and j is from its definition $c^{(i)}$.

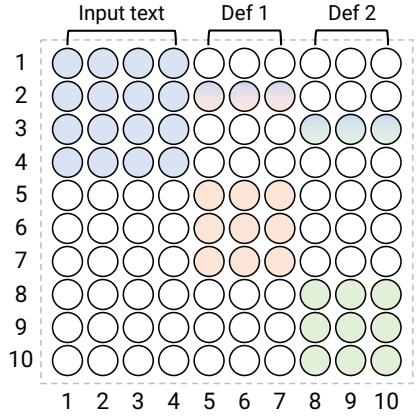


Figure 2: An illustration of knowledge-visible attention matrix. "Def 1" is the dictionary definition of the second word in the input text, and "Def 2" is the definition of the third word in the input text. Colored circle means token i can attention information from token j, while white circle means no attention from token i to token j.

4 Experiments

4.1 Overall Setting

To show the wide adaptability of our Dict-BERT, we conducted experiments on 16 NLP benchmark datasets. we use BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as the backbone pre-trained language methods. First, we followed Liu et al. (2019) and Wu et al. (2021) to use 8 natural language understanding tasks in GLUE, including CoLA, RTE, MRPC, STS, SST, QNLI, QQP, and MNLI. Second, we followed Gururangan et al. (2020) to use 8 specialized domain classification tasks, including Chemprot, RCT-20k, ACL-ARC, SciERC, HyperPartisan, AGNews, Helpfulness, IMDB. The detailed setting of the fine-tuning is illustrated in Appendix A.1.

Table 2: Performance of different models on eight specialized domain datasets under the domain adaptive pre-training (DAPT) setting. Each configuration is run five times with different random seeds, and the average of these five results on the test set is calculated as the final performance.

Methods	ChemProt	RCT	ACL-ARC	SciERC	HP	AGNews	Helpfulness	IMDB	Avg
	Mi-F1	Mi-F1	Ma-F1	Ma-F1	Ma-F1	Ma-F1	Ma-F1	Ma-F1	
BERT	81.16	86.91	64.20	80.40	91.17	94.48	69.39	93.67	82.67
BERT-DAPT	83.10	86.85	71.45	81.62	93.52	**94.58**	70.73	94.78	84.57
Dict-BERT-DAPT-PF	83.49	**87.46**	**74.18**	**83.01**	94.70	**94.58**	70.04	**94.80**	**85.25**
⊢ w/o MIM	83.33	87.38	72.26	82.70	**94.72**	**94.58**	70.33	94.73	85.06
⊢ w/o DD	**84.09**	87.23	72.78	82.54	94.69	94.57	**70.43**	94.70	85.01
RoBERTa	82.03	87.14	66.20	79.55	90.15	94.43	68.35	95.16	83.15
RoBERTa-DAPT	84.02	87.62	73.56	81.85	90.22	94.51	69.06	95.18	84.51
Dict-RoBERTa-PF	84.41	87.42	**75.33**	**82.53**	92.51	**94.80**	70.57	**95.51**	**85.32**
⊢ w/o MIM	**84.49**	**87.51**	74.83	81.58	**93.27**	94.75	70.67	95.40	85.31
⊢ w/o DD	84.09	87.39	74.04	81.18	90.91	94.64	**70.81**	**95.51**	84.82

4.2 Rare Word Collection

Here, we briefly introduce the statistic of rare words in BERT pre-training corpus: English Wikipedia and BookCorpus. By concatenating these two datasets, we obtained a corpus with roughly 16GB in size. The total number of unique words in the pre-training corpus is 504,812, of which 112,750 (22.33%) words are defined as frequent words. In other words, the sum of the occurrences of these 112,750 words in the corpus accounts for 90% of the occurrences of all words in the corpus. We look up definitions of the remaining 392,062 (77.67%) words in the Wiktionary, of which 252581 (50.03%) can be found. The average length of definition is 9.57 words.

4.3 Pre-training Corpus and Tasks

Experiments on the GLUE benchmark: The language model is first pre-trained on the general domain corpus, and then fine-tuned on the training set of different GLUE tasks. Following BERT (Devlin et al., 2019), we used the English Wikipedia and BookCorpus as the pre-training corpus. We removed the next sentence prediction (NSP) as suggested in RoBERTa (Liu et al., 2019), and kept masked language modeling (MLM) as the objective for pre-training a vanilla BERT.

Experiments on specialized domain datasets: The language model is not only pre-trained on the general domain corpus, but also pre-trained on domain specific corpus before fine-tuned on domain specific tasks. To realize it, we initialized our model with the checkpoint from pre-trained BERT/RoBERTa and continue to pre-train on domain-specific corpus (Gururangan et al., 2020). The four domains we focus on are biomedical (BIOMED) papers, computer science (CS) papers, news text from REALNEWS, and e-commerce reviews from AMAZON.

4.4 Baseline Methods

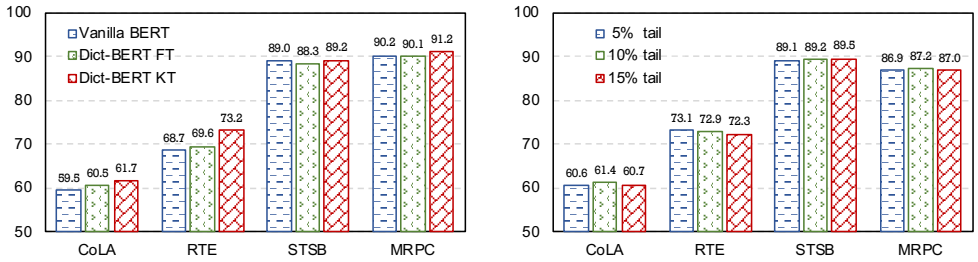
Vanilla BERT/RoBERTa. We use the off-the-shelf BERT-base and RoBERTa-base model and perform supervised fine-tuning of their parameters for each downstream tasks.

BERT-DAPT/RoBERTa-DAPT. It continues pre-training BERT/RoBERTa on a large corpus of unlabeled domain-specific text (e.g., BioMed) using masked language modeling (MLM).

BERT-TNF. It takes notes for rare words on the fly during pre-training to help the model understand them when they occur next time. Specifically, it maintains a note dictionary and saves a rare word's contextual information in it as notes when the rare word occurs in a sentence.

4.5 Ablation Settings

Dict-BERT-F/Dict-BERT-P indicates only using dictionary in the pre-training/fine-tuning stage. Dict-BERT-PF indicates using dictionary in the both pre-training and fine-tuning stage. Dict-BERT w/o MIM removes the word-level mutual information maximization task and Dict-BERT w/o DD removes the sentence-level definition discriminative task.



(a) Full attention v.s. Knowledge attention (b) Rare word ratios (5% v.s. 10% v.s. 15%)

Figure 3: Model performance on CoLA, RTE, STSB and MRPC with different variant settings.

4.6 Evaluation Metrics

For GLUE, we followed RoBERTa (Liu et al., 2019) and reported Matthews correlation for CoLA, Pearson correlation for STS-B, and accuracy for other tasks. For domain adaption, we followed Gururangan et al. (2020) and reported Micro-F1 for Chemprot and RCT-20k, and Macro-F1 for other tasks. For WNLaMPro, we followed Schick & Schütze (2020) and reported MRR and Precision@K.

4.7 Experimental Results

Only using Dictionary during Fine-tuning. As shown in Table 1, comparing with the vanilla BERT and Dict-BERT-F, we can observe that only using dictionary during fine-tuning cannot improve the model performance on the GLUE benchmark. This indicates the pre-trained language model cannot quickly learn rare word definitions in the dictionary to help improve downstream task performance. Furthermore, the pre-trained language model might even be misled by noisy explanations in the dictionary. Therefore, it is important to integrate dictionary into language model pre-training so the dictionary definitions can be better utilized.

Dict-BERT v.s. Baseline Methods. As shown in Table 1, Dict-BERT-PF can outperform the vanilla BERT on the GLUE benchmark by improving +1.15% accuracy on average. The BERT performance from Wu et al. (2021) is higher than our implemented BERT, however, they do not have open-source code for reproducing their experimental results. Though Dict-BERT-PF and BERT-TNF achieved very close performance on GLUE benchmark, i.e., 83.80% and 83.90%, our Dict-BERT-PF has achieved greater relative improvement on the GLUE benchmark than BERT-TNF, i.e., +1.15% and +0.80%. In addition, BERT-TNF keeps a fixed note dictionary so it cannot update any unseen words into the note dictionary during fine-tuning. On the contrary, Dict-BERT can dynamically adjust the vocabulary of rare words, obtain and represent their definitions in dictionary in a plug-and-play manner. On RTE, Dict-BERT-P obtains the biggest performance improvement compared with the vanilla BERT. On another small-data sub-tasks CoLA, Dict-BERT-PF also outperforms the baseline with considerable margins. This indicates that when Dict-BERT is fine-tuned on a small downstream dataset, the improvement is particularly significant. Besides, as shown in Table 2, Dict-BERT-DAPT can outperform BERT-DAPT on the specialized domain datasets by improving +0.68% F1 on average. The same observation can be obtained from the RoBERTa setting.

Fine-tuning with Dictionary v.s. without Dictionary. As shown in Table 1, we compared model performance between using dictionary in fine-tuning and not using dictionary in fine-tuning. First, after pre-training the language model with dictionary, even without using dictionary in fine-tuning, the performance has been greatly improved. This indicates the pre-training the language model with dictionary can generally improve the language representation and provide better initiation before fine-tuning the language model on the downstream tasks. Besides, we also observe the performance of using dictionary can perform slightly better on the GLUE benchmark. We hypothesize the reason behind can be the distribution discrepancy of the pre-training and fine-tuning data.

Knowledge Attention v.s. Full Attention. As we mentioned in the Section 3.4, too much knowledge incorporation may divert the sentence from its original meaning by introducing some noise. This is more likely to happen if there are multiple rare words appeared in the input text. There-

Table 3: Performance of different models on WNLaMPro test set, subdivided by word frequency.

Methods	RARE (0, 10)			FREQUENT (100, +∞)			OVERALL (0, +∞)		
	MRR	P@3	p@10	MRR	P@3	p@10	MRR	P@3	p@10
BERT (base)	0.117	0.053	0.036	0.356	0.179	0.116	0.266	0.130	0.084
Dict-BERT	**0.145**	**0.068**	**0.041**	**0.359**	**0.181**	**0.117**	**0.274**	**0.137**	**0.088**
⊢ w/o MIM	0.144	0.067	**0.041**	0.357	0.180	0.115	0.272	0.135	0.087
⊢ w/o DD	0.141	0.065	0.040	0.355	0.179	0.116	0.269	0.133	0.086

fore, we compared the model performance between using knowledge attention and full attention. Part of experimental results are shown in Figure 3(a) and other results are in Appendix x.x. As shown in Figure 3(a), we observed that using knowledge attention can consistently perform better than using full attention mechanism during the fine-tuning stage on CoLA, RTE, STSB and MRPC datasets. Besides, Dict-BERT with full attention even under-performed than the vanilla BERT without using any dictionary definition, which indicates the appended description in the dictionary may change the meaning of the original sentence. For example, STSB compares similarity between two sentence. Using full attention includes semantic meanings of definitions into the sentence representation, which might reduce the sentence similarity score and hurt the model performance.

Ablation Study. As shown in Table 1 and Table 2, we conducted ablation study on both GLUE benchmark and specialized domain datasets. First, both MIM and DD can help learning knowledge from dictionary and improve language model pre-training. Specifically, DD demonstrates larger average improvement than MIM on two benchmarks. The average improvements on GLUE benchmark brought by DD and MIM are +0.63% and +0.52%. Second, combining MIM and DD together can achieve the highest performance on GLUE benchmark, in which the average gain enlarges to +1.15%. For specialized domain datasets, we have the same observations as above.

Unsupervised Language Model Probing. In order to assess the ability of language models to understand words as a function of their frequency, we used WordNet Language Model Probing (WNLaMPro) dataset (Schick & Schütze, 2020) to test how well a language model understands a given word: we can ask it for properties of that word using natural language. For example, a language model that understands the concept of "guilt", should be able to correctly complete the sentence "Guilt is the opposite of ___" with the word "innocence". WNLaMPro contains four different kinds of relations: antonym, hypernym, cohyponym+, and corruption. Based on the word frequency in English Wikipedia, WNLaMPro defines three subsets based on keyword counts: WNLaMPro-rare $(0, 10)$, WNLaMPro-medium $(10, 100)$, and WNLaMPro-frequent $(100, +\infty)$. As shown in Table 3, Dict-BERT can greatly improve the word representation compared with the vanilla BERT without using a dictionary during pre-training. Based on the word frequency, we observe Dict-BERT can significantly help learn rare word representations. Compared to the vanilla BERT, Dict-BERT improves MRR and P@3 by relatively +23.93% and +28.30%, respectively. In addition, Dict-BERT is also able to learn better frequent word representations. Although we did not directly take frequent word definitions as part of the input, Dict-BERT spends less memory on rare words, because it is easier to predict rare words than the vanilla BERT, so the saved memory power could be used to memorize the facts involving popular words and interactions between popular words.

5 CONCLUSIONS

Enhancing the representation of rare words in language models is an important yet challenging task. To address the rare word problem, in this work, we leveraged rare word definitions in English dictionary to improve rare word representations. During the pre-training stage, when language model encounters a rare word in the input text, we fetch its definition from Wiktionary and append it to the end of the input text. In order to make better interactions between the input text and rare word definitions, we proposed two novel self-supervised training tasks to help language model learn better representations for rare words during the pre-training stage. Experimental on GLUE benchmark and eight specialized domain datasets demonstrate that our method can significantly improve the understanding of rare words and boost model performance on various downstream tasks.

REFERENCES

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 2020.

Leyang Cui, Yu Wu, Shujie Liu, and Yue Zhang. Knowledge enhanced fine-tuning for better handling unseen entities in dialogue generation. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *International Conference for Learning Representation (ICLR)*, 2019.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Frage: frequency-agnostic word representation. In *Conference on Neural Information Processing Systems (Neruips)*, 2018.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. A knowledge-enhanced pretraining model for commonsense story generation. In *Transactions of the Association for Computational Linguistics (TACL)*, 2020.

Beliz Gunel, Chenguang Zhu, Michael Zeng, and Xuedong Huang. Mind the facts: Knowledge-boosted coherent abstractive text summarization. *arXiv preprint arXiv:2006.15435*, 2020.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representation (ICLR)*, 2015.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2020.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Conference on Artificial Intelligence (AAAI)*, 2021.

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

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier, 1989.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning (ICML)*, 2019.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 2020.

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

Elena Sofia Ruzzetti, Leonardo Ranaldi, Michele Mastromattei, Francesca Fallucchi, and Fabio Massimo Zanzotto. Lacking the embedding of a word? look it up into a traditional dictionary. *arXiv preprint arXiv:2109.11763*, 2021.

Timo Schick and Hinrich Schütze. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *International Conference for Learning Representation(ICLR)*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems (Neruips)*, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *International Conference for Learning Representation (ICLR)*, 2019.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021a.

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

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, 2020.

Qiyu Wu, Chen Xing, Yatao Li, Guolin Ke, Di He, and Tie-Yan Liu. Taking notes on the fly helps bert pre-training. *International Conference for Learning Representation (ICLR)*, 2021.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference of Learning Representation (ICLR)*, 2020.

Ruochen Xu, Yuwei Fang, Chenguang Zhu, and Michael Zeng. Does knowledge help general nlu? an empirical study. *arXiv preprint arXiv:2109.00563*, 2021.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems (Neruips)*, 2019.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*, 2020a.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *arXiv preprint arXiv:2010.04389*, 2020b.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, Bill Yuchen Lin, and Xiang Ren. Pre-training text-to-text transformers for concept-centric common sense. In *International Conference for Learning Representation (ICLR)*, 2021.

A APPENDIX

A.1 FINE-TUNING DETAILS

We conduct experiments on pre-training BERT-Base with 110M parameters (Devlin et al., 2019). BERT-base consists of 12 Transformer layers. For each layer, the hidden size is set to 768 and the number of attention head is set to 12. All models are pre-trained for 300k steps with batch size 2,000 and maximum sequence length 512. We use Adam Kingma & Ba (2015) as the optimizer, and set its hyperparameter ϵ to $1e$-6 and (β_1, β_2) to $(0.9, 0.98)$. The peak learning rate is set to $7e$-4. We set the dropout probability to 0.1 and weight decay to 0.01. All configurations are reported in Table 4.

Table 4: Hyperparameters for BERT pre-training and domain-adaptive pre-training (DAPT).

Hyperparameter	**Assignments**	
Pre-training setting	BERT pre-training	Domain adaptive pre-training
number of steps	300K	12.5K
batch size	2,000	2,000
maximum learning rate	7e-4	1e-4
learning rate optimizer	Adam	Adam
Adam epsilon	1e-6	1e-6
Adam beta weights	0.9, 0.98	0.9, 0.98
Weight decay	0.01	0.01
Warmup proportion	0.06	0.06
learning rate decay	linear	linear