Shuō Wén Jiě Zì:
Rethinking Dictionaries and Glyphs for Chinese Language Pre-training

Yuxuan Wang[1,2,3], **Jianghui Wang**[2], **Dongyan Zhao**[1,2,3,4†], **Zilong Zheng**[2,4†]

[1] Wangxuan Institute of Computer Technology, Peking University
[2] Beijing Institute for General Artificial Intelligence (BIGAI)
[3] Center for Data Science, AAIS, Peking University
[4] National Key Laboratory of General Artificial Intelligence
wyx@stu.pku.edu.cn, zhaody@pku.edu.cn, {wangjianghui,zlzheng}@bigai.ai
https://github.com/patrick-tssn/CDBert

Abstract

We introduce CDBERT, a new learning paradigm that enhances the semantics understanding ability of the Chinese Pretrained Language Models (PLMs) with dictionary knowledge and structure of Chinese characters. We name the two core modules of CDBERT as Shuowen and Jiezi, where Shuowen refers to the process of retrieving the most appropriate meaning from Chinese dictionaries and Jiezi refers to the process of enhancing characters' glyph representations with structure understanding. To facilitate dictionary understanding, we propose three pre-training tasks, *i.e.*, Masked Entry Modeling, Contrastive Learning for Synonym and Antonym, and Example Learning. We evaluate our method on both modern Chinese understanding benchmark CLUE and ancient Chinese benchmark CCLUE. Moreover, we propose a new polysemy discrimination task PolyMRC based on the collected dictionary of ancient Chinese. Our paradigm demonstrates consistent improvements on previous Chinese PLMs across all tasks. Moreover, our approach yields significant boosting on few-shot setting of ancient Chinese understanding.

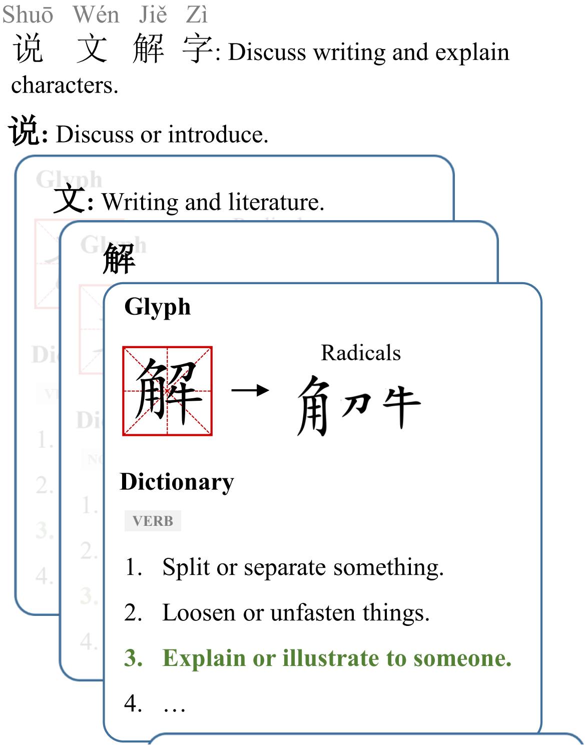
Shuō Wén Jiě Zì
说 文 解 字: Discuss writing and explain characters.
说: Discuss or introduce.
文: Writing and literature.
解
Glyph
Radicals
解 → 角 刀 牛
Dictionary
VERB
1. Split or separate something.
2. Loosen or unfasten things.
3. Explain or illustrate to someone.
4. …

Figure 1: Illustration of CDBERT. The expression in green refers to the selected definition of current character.

1 Introduction

Large-scale pre-trained language models (PLMs) such as BERT (Devlin et al., 2018) and GPT (Brown et al., 2020) have revolutionized various research fields in natural language processing (NLP) landscape, including language generation (Brown et al., 2020), text classification (Wang et al., 2018), language reasoning (Wei et al., 2022), *etc*. The *de facto* paradigm to build such LMs is to feed massive training corpus and datasets to a Transformer-based language model with billions of parameters.

Apart from English PLMs, similar approaches have also been attempted in multilingual (Lample and Conneau, 2019) and Chinese language understanding tasks (Sun et al., 2021b, 2019a). To enhance Chinese character representations, pioneer works have incorporated additional character information, including glyph (character's geometric shape), *pinyin* (character's pronunciation), and stroke (character's writing order) (Sun et al., 2021b; Meng et al., 2019). Nevertheless, there still exists a huge performance gap between concurrent state-of-the-art (SOTA) English PLMs and those on Chinese or other non-Latin languages (Cui et al., 2020), which leads us to rethink the central question: *What are the unique aspects of Chinese that are essential to achieve human-level Chinese understanding?*

With an in-depth investigation of Chinese lan-

[†] Corresponding author: Dongyan Zhao, Zilong Zheng.

Findings of the Association for Computational Linguistics: ACL 2023, pages 1089–1101
July 9-14, 2023 ©2023 Association for Computational Linguistics

guage understanding, this work aims to point out the following crucial challenges that have barely been addressed in previous Chinese PLMs.

- **Frequent *vs*. Rare Characters.** Different from English that enjoys 26 characters to form frequently-used vocabularies (30,522 WordPieces in BERT), the number of frequently-occurred Chinese characters are much smaller (21,128 in Chinese BERT[1]), of which only 3,500 characters are frequently occurred. As of year 2023, over 17 thousand characters have been newly appended to the Chinese character set. Such phenomenon requires models to quickly adapt to rare or even unobserved characters.
- **One *vs*. Many Meanings.** Compared with English expressions, polysemy is more common for Chinese characters, of which most meanings are semantically distinguished. Similar as character set, the meanings of characters keep changing. For example, the character "卷" has recently raised a new meaning: "the involution phenomena caused by peer-pressure".
- **Holistic *vs*. Compositional Glyphs.** Considering the logographic nature of Chinese characters, the glyph information has been incorporated in previous works. However, most work treat glyph as an independent visual image while neglecting its compositional structure and relationship with character's semantic meanings.

In this work, we propose CDBERT, a new Chinese pre-training paradigm that aims to go beyond feature aggregation and resort to mining information from Chinese dictionaries and glyphs' structures, two essential sources that interpret Chinese characters' meaning. We name the two core modules of CDBERT as **Shuowen** and **Jiezi**, in homage to one of the earliest Chinese dictionary in Han Dynasty. Figure 1 depicts the overall model. **Shuowen** refers to the process that finds the most appropriate definition of a character in a Chinese dictionary. Indeed, resorting to dictionaries for Chinese understanding is not unusual even for Chinese Linguistic experts, especially when it comes to ancient Chinese (*aka.* classical Chinese) understanding. Different from previous works that simply use dictionaries as additional text corpus (Yu et al., 2021; Chen et al., 2022), we propose a fine-grained definition retrieval framework from Chinese dictionaries. Specifically, we design three types of objectives for dictionary pre-training: Masked Entry Modeling (MEM) to learn entry representation; Contrastive Learning objective with synonyms and antonyms; Example Leaning (EL) to distinguish polysemy by example in the dictionary. **Jiezi** refers to the process of decomposing and understanding the semantic information existing in the glyph information. Such a process grants native Chinese the capability of understanding new characters. In CDBERT, we leverage radical embeddings and previous success of CLIP (Yang et al., 2022; Radford et al., 2021) model to enhance model's glyph understanding capability.

We evaluate CDBERT with extensive experiments and demonstrate consistent improvements of previous baselines on both modern Chinese and ancient Chinese understanding benchmarks. It is worth noting that our method gets significant improvement on CCLUE-MRC task in few-shot setting. Additionally, we construct a new dataset aiming to test models' ability to distinguish polysemy in Chinese. Based on the $BaiduHanyu$, we construct a polysemy machine reading comprehension task (PolyMRC). Given the example and entry, the model needs to choose a proper definition from multiple interpretations of the entry. We believe our benchmark will help the development of Chinese semantics understanding.

In summary, the contributions of this work are four-fold: (i) We propose CDBERT, a new learning paradigm for improving PLMs with Chinese dictionary and characters' glyph representation; (ii) We derive three pre-training tasks, Masked Entry Modeling, Contrastive Learning for Synonym and Antonym, and Example Learning, for learning a dictionary knowledge base with a polysemy retriever (Sec. 3.1); (iii) We propose a new task PolyMRC, specially designed for benchmarking model's ability on distinguishing polysemy in ancient Chinese. This new task complements existing benchmarks for Chinese semantics understanding (Sec. 4); (iv) We systematically evaluate and analyze the CDBERT on both modern Chinese and ancient Chinese NLP tasks, and demonstrate improvements across all these tasks among different types of PLMs. In particular, we obtain significant performance boost for few-shot setting in ancient Chinese understanding.

2 Related Work

Chinese Language Model Chinese characters, different from Latin letters, are generally lo-

[1] https://github.com/ymcui/Chinese-BERT-wwm

grams. At an early stage, Devlin et al. (2018); Liu et al. (2019b) propose BERT-like language models with character-level masking strategy on Chinese corpus. Sun et al. (2019b) take phrase-level and entity-level masking strategies to learn multi-granularity semantics for PLM. Cui et al. (2019) pre-trained transformers by masking all characters within a Chinese word. Lai et al. (2021) learn multi-granularity information with a constructed lattice graph. Recently, Zhang et al. (2020); Zeng et al. (2021); Su et al. (2022b) pre-trained billion-scale parameters large language models for Chinese understanding and generation. In addition to improving masking strategies or model size, some researchers probe the semantics from the structure of Chinese characters to enhance the word embedding. Since Chinese characters are composed of radicals, components, and strokes hierarchically, various works (Sun et al., 2014; Shi et al., 2015; Li et al., 2015; Yin et al., 2016; Xu et al., 2016; Ma et al., 2020; Lu et al., 2022) learn the Chinese word embedding through combining indexed radical embedding or hierarchical graph. Benefiting from the strong representation capability of convolutional neural networks (CNNs), some researchers try to learn the morphological information directly from the glyph (Liu et al., 2017; Zhang and LeCun, 2017; Dai and Cai, 2017; Su and yi Lee, 2017; Tao et al., 2019; Wu et al., 2019). Sehanobish and Song (2020); Xuan et al. (2020) apply the glyph-embedding to improve the performance of BERT on named entity recognition (NER). Besides, polysemy is common among Chinese characters, where one character may correspond to different meanings with different pronunciations. Therefore, Zhang et al. (2019) use “pinyin” to assist modeling in distinguishing Chinese words. Sun et al. (2021c) first incorporate glyph and “pinyin” of Chinese characters into PLM, and achieve SOTA performances across a wide range of Chinese NLP tasks. Su et al. (2022a) pre-trained a robust Chinese BERT with synthesized adversarial contrastive learning examples including semantic, phonetic, and visual features.

Knowledge Augmented pre-training Although PLMs have shown great success on many NLP tasks. There are many limitations on reasoning tasks and domain-specific tasks, where the data of downstream tasks vary from training corpus in distribution. Even for the strongest LLM ChatGPT, which achieves significant performance boost across a wide range of NLP tasks, it is not able to answer questions involving up-to-date knowledge. And it is impossible to train LLMs frequently due to the terrifying costs. As a result, researchers have been dedicated to injecting various types of knowledge into PLM/LLM. According to the types, knowledge in existing methods can be classified to text knowledge (Hu et al., 2022) and graph knowledge, where text knowledge can be further divided into linguistic knowledge and non-linguistic knowledge. Specifically, some works took lexical information (Lauscher et al., 2019; Zhou et al., 2020; Lyu et al., 2021) or syntax tree (Sachan et al., 2020; Li et al., 2020; Bai et al., 2021) to enhance the ability of PLMs in linguistic tasks. For the non-linguistic knowledge, some researchers incorporate general knowledge such as Wikipedia with retrieval methods (Guu et al., 2020; Yao et al., 2022; Wang et al., 2022) to improve the performance on downstream tasks, others use domain-specific corpora (Lee et al., 2019; Beltagy et al., 2019) to transfer the PLMs to corresponding downstream tasks. Compared with text knowledge, a knowledge graph contains more structured information and is better for reasoning. Thus a flourish of work (Liu et al., 2019a; Yu et al., 2020; He et al., 2021; Sun et al., 2021a; Zhang et al., 2022) designed fusion methods to combine the KG with PLMs.

Dictionary Augmented pre-training Considering the heavy-tailed distribution of the pre-training corpus and difficult access to the knowledge graph, some works injected dictionary knowledge into PLMs to alleviate the above problems. (Yu et al., 2021) enhance PLM with rare word definitions from English dictionaries. Chen et al. (2022) pre-trained BERT with English dictionary as a pre-training corpus and adopt an attention-based infusion mechanism for downstream tasks.

3 CDBert

3.1 Shuowen: Dictionary as a pre-trained Knowledge

We take three steps while looking up the dictionary as the pre-training tasks: 1) Masked Entry Modeling (MEM). The basic usage of a dictionary is to clarify the meaning of the entry. 2) Contrastive Learning for Synonym and Antonym (CL4SA). For ambiguous meanings, we always refer to the synonym and antonym for further understanding. 3) Example Learning (EL). We will figure out the accurate meaning through several classical examples.

Masked Entry Modeling (MEM) Following existing transformer-based language pre-training models (Devlin et al., 2018; Liu et al., 2019b), we take the MEM as a pre-training task. Specifically, we concatenate the entry (`<ent>`) with its corresponding meaning or definition (`<def>`) as input, *i.e.*, {`[CLS] <ent> [SEP] <def> [SEP]`}. Then the MEM task masks out the `<ent>` with a `[MASK]` token, and attempts to recover it. Considering the entry might be composed of multiple characters, we use whole word masking (WWM) (Cui et al., 2020) as the entry masking strategy. The objective of MEM $\mathcal{L}_{mem}$ is computed as the cross-entropy between the recovered entry and the ground truth.

Contrastive Learning for Synonym and Antonym (CL4SA) Inspired by Yang et al. (2022), we adopt contrastive learning to better support the semantics of the pre-trained representation. We construct positive sample pair $\langle ent, syno\rangle$ with synonyms in the dictionary, and negative sample pair $\langle ent, anto\rangle$ with antonyms in the dictionary. The goal of the CL4SA is to make the positive sample pair closer while pushing the negative sample pair further. Thus we describe the contrastive objective as follows:

$$\mathcal{L}_{cl4sa} = -\log \frac{e^{h_{ent} \cdot h_{syno}}}{e^{h_{ent} \cdot h_{syno}} + e^{h_{ent} \cdot h_{anto}}}$$

where $\cdot$ denotes the element-wise product, h_{ent}, h_{syno}, h_{anto} is the representation of the original entry, the synonym, and the antonym respectively. In practice, we use the hidden states of `[CLS]` token as the representation of the input {`[CLS]<ent>[SEP]<def>[SEP]`}. Since the antonyms in the dictionary are much less than synonyms, we randomly sampled entries from the vocabulary for compensation. To distinguish the sampled entries with the strict antonyms, we set different weights for them.

Example Learning (EL) Compared with other languages, the phenomenon of polysemy in Chinese is more serious, and most characters or words have more than one meanings or definitions. To better distinguish multiple definitions of an entry in a certain context, we introduce example learning, which attempts to learn the weight of different definitions for a certain example. Specifically, given an entry ent, K multiple definitions $def_1, \ldots, def_K$, and an exemplar phrase exa_i of meaning def_i, we use h_{exa}, the hidden state of the `[CLS]` token in the example as query Q, and $X = \{h^i_m\}^k_{i=1}$, the hidden states of the `[CLS]` token in the meanings as key K. Then the attention score can be computed as:

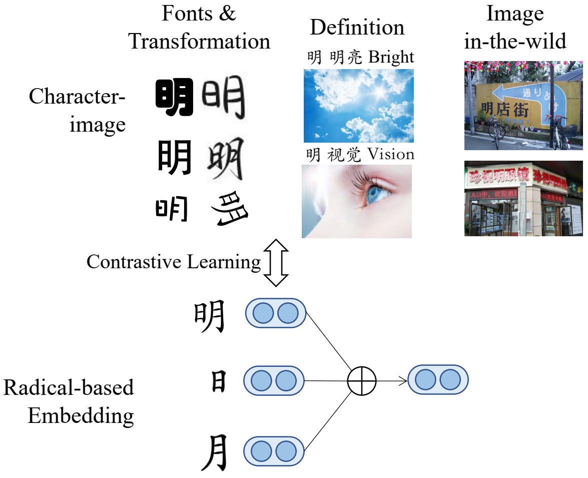

Figure 2: Illustration of glyph-enhanced character representation on character "明".

$$Attn_{def} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{1}$$

We use the cross-entropy loss to supervise the meaning of retriever training:

$$\mathcal{L}_{el} = CrossEntropy(\text{one-hot}(def), Attn_{def}) \tag{2}$$

where one-hot$(\cdot)$ is a one-hot vector transition of ground-truth indexes.

We sum over all the above objectives to obtain the final loss function:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{mem} + \lambda_2 \mathcal{L}_{cl4sa} + \lambda_3 \mathcal{L}_{el} \tag{3}$$

where λ_1, λ_2, λ_3 are three hyper-parameters to balance three tasks.

3.2 Jiezi: Glyph-enhanced Character Representation

Chinese characters, different from Latin script, demonstrate strong semantic meanings. We conduct two structured learning strategies to capture the semantics of Chinese characters. Following Sun et al. (2021b), we extract the glyph feature by the CNN-based network.

CLIP enhanced glyph representation To better capture the semantics of glyphs, we learn the glyph representation through a contrastive learning algorithm. Specifically, we concatenate character c with its definition def as text input and generate a picture of the character as visual input. We initialize our model with the pre-trained checkpoint of Chinese-CLIP (Yang et al., 2022) and keep the symmetric cross-entropy loss over the similarity scores between text input and visual input as objectives. To alleviate the influence of pixel-level noise, we follow Jaderberg et al. (2014, 2016) to generate a large number of images of characters by transformation, including font, size, direction, *etc*. Besides, we introduce some Chinese character images in wild (Yuan et al., 2019) in the training corpus to improve model robustness. Finally, we extract the glyph feature through the text encoder to mitigate the pixel bias.

Radical-based character embedding Since the glyph feature requires extra processing and is constrained by the noise in images, we propose a radical-based embedding for end-to-end pre-training. We first construct a radical vocabulary, then add the radical embedding for each character with their radical token in the radical vocabulary. Thus, we can pre-train the CDBERT in the end-to-end learning method.

3.3 Applying CDBERT to downstream tasks

Following Chen et al. (2022), we use the CDBERT as a knowledge base for retrieving entry definitions. Specifically, given an input expression, we first look up all the entries in the dictionary. Then, we adopt the dictionary pre-training to get the representation of the entry. At last, we fuse the CDBERT-augmented representation to the output of the language model for further processing in downstream tasks. We take the attention block pre-trained by the EL task as a retriever to learn the weight of all the input entries with multiple meanings. After that, we use weighted sum as a pooling strategy to get the CDBERT-augmented representation of the input. We concatenate the original output of the language model with the CDBERT-augmented representation for final prediction.

4 The PolyMRC Task

Most existing Chinese language understanding evaluation benchmarks do not require the model to have strong semantics understanding ability. Hence, we propose a new dataset and a new machine reading comprehension task focusing on polysemy understanding. Specifically, we construct a dataset through entries with multiple meanings and examples from dictionaries. As for the Polysemy Machine Reading Comprehension (PolyMRC) task, we set the example as context and explanations as choices, the goal of PolyMRC is to find the correct explanation of the entry in the example. Table 1 shows the statistics of the dataset.

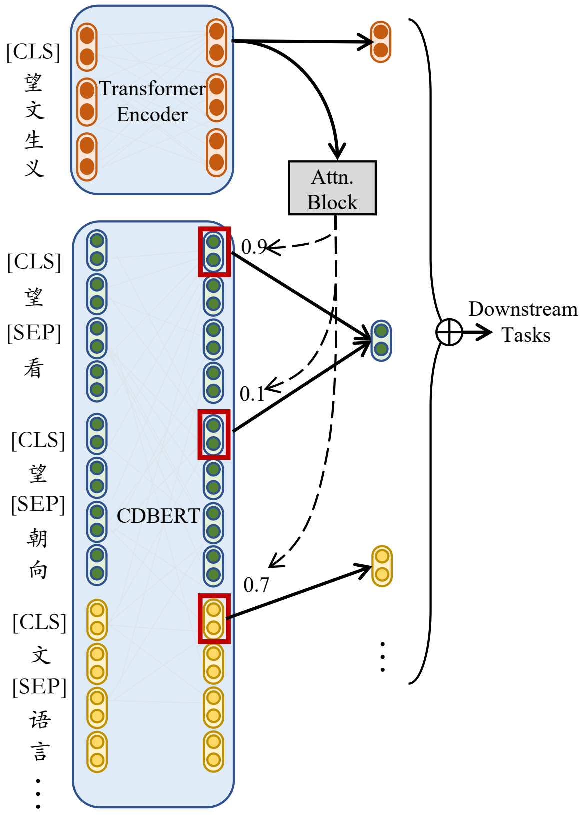

Figure 3: Illustration of applying CDBERT to downstream tasks. ⊕ indicates the concatenation operation. The Attn. Block is the pre-trained attention model from the EL task.

Table 1: Statistics of PolyMRC Dataset

Split	Sentences	Average length
Training data	46,119	38.55
Validation data	5,765	38.31
Test data	5,765	38.84

5 Experiments

5.1 Implementation Details

We pre-train CDBert based on multiple official pre-trained Chinese BERT models. All the models are pre-trained for 10 epochs with batch size 64 and maximum sequence length 256. We adopt AdamW as the optimizer and set the learning rate as $5e-5$ with a warmup ratio of 0.05. We set $\lambda_1 = 0.6$, $\lambda_2 = 0.2$, and $\lambda_3 = 0.2$ in Eqn. (3) for all the experiments. We finetune CLUE (Xu et al., 2020) with the default setting reported in the CLUE GitHub repository[2].

5.2 Baselines

BERT We adopt the official BERT-base model pre-trained on the Chinese Wikipedia corpus as baseline models.

RoBERTa Besides BERT, we use two stronger PLMs as baselines: RoBERTa-base-wwm-ext and RoBERTa-large-wwm-ext (we will use RoBERTa and RoBERTa-large for simplicity). In these models, *wwm* denotes the model continues pre-training on official RoBERTa models with the WWM strategy, and *ext* denotes the models are pre-trained on extended data besides Wikipedia corpus.

MacBERT MacBERT improves on RoBERTa by taking the MLM-as-correlation (MAC) strategy and adding sentence ordering prediction (SOP) as a new pre-training task. We use MacBERT-large as a strong baseline method.

5.3 CLUE

We evaluate the general natural language understanding (NLU) capability of our method with CLUE benchmark (Xu et al., 2020), which includes text classification and machine reading comprehension (MRC) tasks. There are five datasets for text classification tasks: **CMNLI** for natural language inference, **IFLYTEK** for long text classification, **TNEWS'** for short text classification, **AFQMC** for semantic similarity, **CLUEWSC 2020** for coreference resolution, and **CSL** for keyword recognition. The text classification tasks can further be classified into single-sentence tasks and sentence pair tasks. The MRC tasks include span selection-based **CMRC2018**, multiple choice questions C3, and idiom Cloze ChID.

The results of text classification are shown in Table 2. In general, CDBERT performs better on single-sentence tasks than sentence pair tasks. Specifically, compared with baselines, CDBERT achieves an average improvement of 1.8% on single sentence classification: TNEWS', IFLYTEK, and WSC. Besides, CDBERT outperforms baselines on long text classification task IFLYTEK by improving 2.08% accuracy on average, which is more significant than the results (1.07%) on short text classification task TNEWS'. This is because TNEWS' consists of news titles in 15 categories, and most titles consist of common words which are easy to understand. But IFLYTEK is a long text 119 classification task that requires comprehensive understanding of the context. In comparison, the average improvement on sentence pair tasks brought by CDBERT is 0.76%, which is worse than the results on single sentence tasks. These results show dictionary is limited in helping PLM to improve the ability of advanced NLU tasks, such as sentiment entailment, keywords extraction, and natural language inference.

We demonstrate the results on MRC tasks in Table 3. As we can see, CDBERT yields a performance boost on MRC tasks (0.79%) on average among all the baselines. It is worth noting that when the PLM gets larger in parameters and training corpus, the gain obtained by CDBERT becomes less. We believe this is caused by the limitation of CLUE benchmark for the reason that several large language models have passed the performance of humans (Xu et al., 2020).

5.4 CCLUE

Ancient Chinese (*aka.* Classical Chinese) is the essence of Chinese culture, but there are many differences between ancient Chinese and modern Chinese. CCLUE[3] is a general ancient NLU evaluation benchmark including NER task, short sentence classification task, long sentence classification task, and machine reading comprehension task. We use the CCLUE benchmark to evaluate the ability of CDBERT to adapt modern Chinese pre-trained models to ancient Chinese understanding tasks.

In order to assess the ability of modern Chinese PLM to understand ancient Chinese by CDBERT, we test our model on CCLUE benchmark. We pretrain the CDBERT on the ancient Chinese dictionary for fairness. Results are presented in Table 4, which shows CDBERT is helpful in all three general NLU tasks: sequence labeling, text classification, and machine reading comprehension. We find in MRC task, CDBERT improves from 42.93 on average accuracy of all 4 models to 44.72 (4.15% relatively), which is significantly better than other tasks. In addition, we can see the gain obtained from the model scale is less than CDBERT on CLUE datasets. This is because the training corpus

[2] https://github.com/CLUEbenchmark/CLUE

[3] https://cclue.top

Table 2: Performance improvements of CDBERT on CLUE$_{classification}$.

Model	AFQMC	TNEWS'	IFLYTEK	CMNLI	WSC	CSL	SCORE
BERT$_{\text{Base}}$	73.70	56.58	60.29	79.69	70	80.36	70.10
BERT$_{\text{Base}}$+ CDBERT	73.48	57.19	62.12	80.19	71.38	81.4	70.96
RoBERTa$_{\text{ext}}$	74.04	56.94	60.31	80.51	80.69	81	72.25
RoBERTa$_{\text{ext}}$+ CDBERT	74.88	57.68	62.19	81.81	81.38	80.93	73.15
RoBERTa$_{\text{ext-large}}$	76.55	58.61	62.98	82.12	82.07	82.13	74.08
RoBERTa$_{\text{ext-large}}$+ CDBERT	**76.82**	**59.09**	**63.04**	**82.89**	**84.83**	**83.07**	**74.95**

Table 3: Performance of CDBERT on CLUE$_{MRC\&QA}$. * we can not reproduce the result reported in CLUE github repo. (<report in github repo>-<report in paper>)

Model	CMRC2018	CHID	C3	SCORE
BERT$_{\text{Base}}$	71.6	80.04	64.50	72.71
BERT$_{\text{Base}}$+ CDBERT	71.75	82.61	65.39	73.25
RoBERTa$_{\text{ext}}$	75.20	83.62	66.50	75.11
RoBERTa$_{\text{ext}}$+ CDBERT	75.85	84.7	67.09	75.88
RoBERTa*$_{\text{ext-large}}$	76.65 (77.95-76.58)	85.32 (85.37-85.37)	73.72 (73.82-72.32)	78.56
RoBERTa$_{\text{ext-large}}$+ CDBERT	77.75	85.38	73.95	79.03

Table 4: Performance of CDBERT on CCLUE.

Model	NER	CLS	SENT	MRC	SCORE
BERT$_{\text{Base}}$	71.62	82.31	59.95	42.76	64.16
BERT$_{\text{Base}}$+ CDBERT	72.41	82.74	60.25	43.91	64.83
RoBERTa$_{\text{ext}}$	69.5	81.96	59.4	42.3	63.29
RoBERTa$_{\text{ext}}$+CDBERT	70.89	82.15	59.95	44.14	64.28
RoBERTa$_{\text{ext-large}}$	79.87	82.9	58.4	43.45	66.16
RoBERTa$_{\text{ext-large}}$+ CDBERT	79.93	83.03	59.75	45.52	67.06
MacBERT$_{\text{ext-large}}$	81.89	83.06	58.9	43.22	66.77
MacBERT$_{\text{ext-large}}$+ CDBERT	82.33	83.71	59.4	45.29	67.68

Table 5: Performance on PolyMRC.

Model	Accuracy
BERT$_{\text{Base}}$	65.33
BERT$_{\text{Base}}$+ CDBERT	65.93
RoBERTa$_{\text{ext}}$	61.96
RoBERTa$_{\text{ext}}$+ CDBERT	62.93
RoBERTa$_{\text{ext-large}}$	64.18
RoBERTa$_{\text{ext-large}}$+ CDBERT	64.77
MacBERT$_{\text{ext-large}}$	66.73
MacBERT$_{\text{ext-large}}$+ CDBERT	67.16

of these PLMs do not contain ancient Chinese. In this scenario, CDBERT is more robust.

PolyMRC Results We use BERT, RoBERTa, and MacBERT as baselines for the new task. Considering the context of PolyMRC is examples in dictionary, we carefully filter out the entries in test set from pre-training corpus, and only take the MEM and CL4SA as pre-training tasks. The results are shown in Table 5. Compared to baselines, CDBERT shows a 1.01% improvement for accuracy on average. We notice that the overall performance show weak relation with the scale of the training corpus of PLM, which is a good sign as it reveals that the new task can not be solved by models simply adding training data.

5.5 FewShot Setting on PolyMRC and CCLUE-MRC

To further investigate the ability of CDBERT on few-shot setting, we construct two challenge datasets based on CCLUE MRC and PolyMRC.

Table 6: Performance of CDBERT on 10-shot setting of two MRC benchmarks.

Model	PolyMRC	CCLUE-MRC
BERT$_{\text{Base}}$	30.98	23.68
BERT$_{\text{Base}}$+ CDBERT	36.65	28.05
RoBERTa$_{\text{ext}}$	28.85	26.67
RoBERTa$_{\text{ext}}$+ CDBERT	29.47	28.51
RoBERTa$_{\text{ext-large}}$	28.45	25.06
RoBERTa$_{\text{ext-large}}$+ CDBERT	29.35	27.59
MacBERT$_{\text{ext-large}}$	37.35	25.29
MacBERT$_{\text{ext-large}}$+ CDBERT	39.22	27.81

Following Few CLUE benchmark [few CLUE], we collect 10 samples for these two MRC tasks. Additionally, we build three different training samples to alleviate the possible fluctuating results of models training on small datasets. We demonstrate the results on Table 6. Compared with BERT, CDBERT+BERT improves on accuracy from 30.98 to 36.65 (18.3% relatively) on PolyMRC, from 23.68 to 28.05 (18.45% relatively) on CCLUE-MRC. The performance gain on BERT is much more significant than larger baselines. This observation indicates that CDBERT is promising in semantics understanding with a handful of annotated training data.

5.6 Ablation Study

Table 7: Ablation of CDBERT on CCLUE-MRC.

Model	ACC
RoBERTa+CDBERT	44.14
RoBERTa	42.30
- Radical	43.68
Replace with Glyph	42.53
Replace with Char Dict	42.76
w/o. CL4SA	43.68
w/o. EL	42.99
Continuous-pre-train	43.14

We conduct ablation studies on different components of CDBERT. We use the CCLUE-MRC for analysis and take the Roberta$_{base}$ as the backbone. The overall results are shown in Table 7. Generally, CDBERT improves the Roberta from 42.30 to 44.14 (4.3% relatively).

The Effect of Character Structure We first evaluate the effects of radical embeddings and glyph embeddings. For fair comparisons, we keep other settings unchanged, and focus on the following setups: "-Radical", where radical embedding is not considered; "Rep Glyph", where we replace the radical embedding with glyph embedding. Results are shown in row 3-4. As can be seen, when we replace the radical embedding with glyph embedding, the accuracy drops 1.61 points, where the performance degradation is more obvious than removing radical embedding. The reason we use here is the scale of training corpus is not large enough to fuse the pre-trained glyph feature to CDBERT.

The Effect of Dictionary We then assess the effectiveness of the dictionary. We replace the original dictionary with character dictionary (row 5) and keep the model size and related hyper-parameters the same as CDBERT pre-training procedure for fair. Besides, during finetuning process, we identify all the characters that are included in the character dictionary for further injecting with dictionary knowledge. We observe the character CDBERT is helpful to some degree (1.1%) but is much worse than the original CDBERT. On the one hand, the number of characters in Chinese is limited, on the other hand, a word and its constituent characters may have totally different explanations.

The Effect of Pre-training Tasks At last, we evaluate different pre-training tasks of CDBERT including CL4SA and EL (row 6-7). Specifically, both CL4SA and EL help improve the NLU ability of PLM, and EL demonstrated larger improvement than CL4SA. The average improvements on CCLUE-MRC brought by CL4SA and EL are 1.05% and 2.68%. In order to verify the impact of CDBERT instead of the additional corpus, we follow Cui et al. (2019) to continuously pre-train the Roberta on the dictionary, which is regarded as extended data. As shown in row 8, using additional pre-training data results in further improvement. However, such improvement is less than our proposed CDBERT, which is a drop of 1 point.

6 Limitations

We collect the dictionary from the Internet, and although we make effort to reduce replicate explanations, there is noise in the dictionary. Besides, not all the words are included in the dictionary. In other words, the quality and amount of entries in the Chinese dictionary are to be improved. Additionally, our method is pre-trained on the Bert-like transformers to enhance the corresponding PLMs, and can not be applied to LLM directly whose frameworks are unavailable. In the future, we will use the retriever for disambiguation and dictionary knowledge infusion to LLM.

7 Conclusion

In this work, we leverage Chinese dictionary and structure information of Chinese characters to enhance the semantics understanding ability of PLM. To make Chinese Dictionary knowledge better act on PLM, we propose 3 pre-training objectives simulating looking up dictionary in our study, and incorporate radical or glyph features to CDBERT. Experiment results on both modern Chinese tasks and ancient Chinese tasks show our method significantly improve the semantic understanding ability of various PLM. In the future, we will explore our method on more high-quality dictionaries (*e.g.* Bilingual dictionary), and adapt our method to LLM to lessen the semantic errors. Besides, we will probe more fine-grained structure information of logograms in both understanding and generation tasks.

Acknowledgements

This project is supported by National Key R&D Program of China (2021ZD0150200).

References

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, J. Yu, and Yunhai Tong. 2021. Syntax-bert: Improving pre-trained transformers with syntax trees. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901.

Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang. 2022. Dictbert: Dictionary description knowledge enhanced language model pre-training via contrastive learning. *arXiv preprint arXiv:2208.00635*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Falcon Z. Dai and Zheng Cai. 2017. Glyph-aware embedding of chinese characters. In *SWCN@EMNLP*.

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*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning (ICML)*.

Lei He, Suncong Zheng, Tao Yang, and Feng Zhang. 2021. Klmo: Knowledge graph enhanced pretrained language model with fine-grained relationships. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2022. A survey of knowledge-enhanced pre-trained language models. *ArXiv*, abs/2212.13428.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NIPS*.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2016. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision (IJCV)*, 116(1):1–20.

Yuxuan Lai, Yijia Liu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2021. Lattice-bert: Leveraging multi-granularity representations in chinese pre-trained language models. *NAACL*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Anne Lauscher, Ivan Vulic, E. Ponti, Anna Korhonen, and Goran Glavavs. 2019. Specializing unsupervised pretraining models for word-level semantic similarity. In *International Conference on Computational Linguistics (COLING)*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36:1234 – 1240.

Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. 2015. Component-enhanced chinese character embeddings. *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2020. Improving bert with syntax-aware local attention. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning character-level compositionality with visual features. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-bert: Enabling language representation with knowledge graph. In *AAAI Conference on Artificial Intelligence (AAAI)*.

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

Wei Lu, Zhaobo Zhang, Pingpeng Yuan, Hai rong Jin, and Qiangsheng Hua. 2022. Learning chinese word embeddings by discovering inherent semantic relevance in sub-characters. *Proceedings of the 31st*

ACM International Conference on Information & Knowledge Management.

Bo Lyu, Lu Chen, Su Zhu, and Kai Yu. 2021. Let: Linguistic knowledge enhanced graph transformer for chinese short text matching. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Bing Ma, Q. Qi, Jianxin Liao, Haifeng Sun, and Jingyu Wang. 2020. Learning chinese word embeddings from character structural information. *Comput. Speech Lang.*, 60.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. *Advances in Neural Information Processing Systems*, 32.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR.

Devendra Singh Sachan, Yuhao Zhang, Peng Qi, and William Hamilton. 2020. Do syntax trees help pre-trained transformers extract information? In *Conference of the European Chapter of the Association for Computational Linguistics*.

Arijit Sehanobish and Chan Hee Song. 2020. Using chinese glyphs for named entity recognition. *AAAI Conference on Artificial Intelligence (AAAI)*.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. 2015. Radical embedding: Delving deeper to chinese radicals. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hui Su, Weiwei Shi, Xiaoyu Shen, Zhou Xiao, Tuo Ji, Jiarui Fang, and Jie Zhou. 2022a. RoCBert: Robust Chinese bert with multimodal contrastive pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 921–931, Dublin, Ireland. Association for Computational Linguistics.

Hui Su, Xiao Zhou, Houjin Yu, Yuwen Chen, Zilin Zhu, Yang Yu, and Jie Zhou. 2022b. Welm: A well-read pre-trained language model for chinese. *ArXiv*.

Tzu-Ray Su and Hung yi Lee. 2017. Learning chinese word representations from glyphs of characters. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019a. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019b. Ernie: Enhanced representation through knowledge integration. *ACL*.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2021a. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021b. ChineseBERT: Chinese pretraining enhanced by glyph and Pinyin information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2065–2075, Online. Association for Computational Linguistics.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021c. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information.

Hanqing Tao, Shiwei Tong, Tong Xu, Qi Liu, and Enhong Chen. 2019. Chinese embedding via stroke and glyph information: A dual-channel view. *ArXiv*, abs/1906.04287.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Shuo Wang, Yichong Xu, Yuwei Fang, Yang Liu, S. Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*.

Wei Wu, Yuxian Meng, Fei Wang, Qinghong Han, Muyu Li, Xiaoya Li, Jie Mei, Ping Nie, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jian Xu, Jiawei Liu, Liangang Zhang, Zhengyu Li, and Huanhuan Chen. 2016. Improve chinese word embeddings by exploiting internal structure. In *North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhenyu Xuan, Rui Bao, Chuyu Ma, and Shengyi Jiang. 2020. Fgn: Fusion glyph network for chinese named entity recognition. In *China Conference on Knowledge Graph and Semantic Computing*.

An Yang, Junshu Pan, Junyang Lin, Rui Men, Yichang Zhang, Jingren Zhou, and Chang Zhou. 2022. Chinese clip: Contrastive vision-language pretraining in chinese. *arXiv preprint arXiv:2211.01335*.

Yunzhi Yao, Shaohan Huang, Ningyu Zhang, Li Dong, Furu Wei, and Huajun Chen. 2022. Kformer: Knowledge injection in transformer feed-forward layers. In *Natural Language Processing and Chinese Computing*.

Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. 2016. Multi-granularity chinese word embedding. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2020. Jaket: Joint pre-training of knowledge graph and language understanding. In *AAAI Conference on Artificial Intelligence (AAAI)*.

W. Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2021. Dict-bert: Enhancing language model pre-training with dictionary.

Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, Tai-Jiang Mu, and Shi-Min Hu. 2019. A large chinese text dataset in the wild. *Journal of Computer Science and Technology*, 34(3):509–521.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, Chen Li, Ziyan Gong, Yifan Yao, Xinjing Huang, Jun Wang, Jia xin Yu, Qiwei Guo, Yue Yu, Yan Zhang, Jin Wang, Heng Tao, Dasen Yan, Zexuan Yi, Fang Peng, Fan Jiang, Han Zhang, Lingfeng Deng, Yehong Zhang, Zhengping Lin, Chao Zhang, Shaojie Zhang, Mingyue Guo, Shanzhi Gu, Gaojun Fan, Yaowei Wang, Xuefeng Jin, Qun Liu, and Yonghong Tian. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *ArXiv*, abs/2104.12369.

Xiang Zhang and Yann LeCun. 2017. Which encoding is the best for text classification in chinese, english, japanese and korean? *ArXiv*, abs/1708.02657.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models. In *International Conference on Learning Representations (ICLR)*.

Yun Zhang, Yongguo Liu, Jiajing Zhu, Ziqiang Zheng, Xiaofeng Liu, Weiguang Wang, Zijie Chen, and Shuangqing Zhai. 2019. Learning chinese word embeddings from stroke, structure and pinyin of characters. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, S. Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juan-Zi Li, Xiaoyan Zhu, and Maosong Sun. 2020. Cpm: A large-scale generative chinese pre-trained language model. *AI Open*.

Junru Zhou, Zhuosheng Zhang, and Hai Zhao. 2020. Limit-bert : Linguistics informed multi-task bert. *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
limitation section

☑ A2. Did you discuss any potential risks of your work?
limitation section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract section and 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3,4

☑ B1. Did you cite the creators of artifacts you used?
3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
4

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
4

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.