EasyNLP: A Comprehensive and Easy-to-use Toolkit for Natural Language Processing

Chengyu Wang¹, Minghui Qiu¹∗, Taolin Zhang¹,², Tingting Liu¹,², Lei Li¹,², Jianing Wang¹,², Ming Wang¹, Jun Huang¹, Wei Lin¹
¹ Platform of AI (PAI), Alibaba Group ² East China Normal University
{chengyu.wcy,minghui.qmh,huangjun.hj}@alibaba-inc.com

Abstract

Pre-Trained Models (PTMs) have reshaped the development of Natural Language Processing (NLP) and achieved significant improvement in various benchmarks. Yet, it is not easy for industrial practitioners to obtain high-performing PTM-based models without a large amount of labeled training data and deploy them online with fast inference speed. To bridge this gap, EasyNLP is designed to make it easy to build NLP applications, which supports a comprehensive suite of NLP algorithms. It further features knowledge-enhanced pre-training, knowledge distillation and few-shot learning functionalities, and provides a unified framework of model training, inference and deployment for real-world applications. EasyNLP has powered over ten business units within Alibaba Group and is seamlessly integrated to the Platform of AI (PAI) products on Alibaba Cloud. The source code of EasyNLP is released at GitHub (https://github.com/alibaba/EasyNLP).

1 Introduction

Pre-Trained Models (PTMs) such as BERT, GPT-3 and PaLM have achieved remarkable results in NLP. With the scale expansion of PTMs, the performance of NLP tasks has been continuously improved; thus, there is a growing trend of ultra-large-scale pre-training, pushing the scale of PTMs from millions, billions, to even trillions (Devlin et al., 2019; Brown et al., 2020; Chowdhery et al., 2022).

However, the application of large PTMs in industrial scenarios is still a non-trivial problem, with reasons as follows. i) Large PTMs are not always smarter and can make commonsense mistakes due to the lack of world knowledge (Petroni et al., 2019). Hence, it is highly necessary to make PTMs explicitly understand world facts by knowledge-enhanced pre-training, especially for supporting domain-specific applications. ii) Although large-scale PTMs have achieved good results with few training samples, the problem of insufficient data and the huge size of models such as GPT-3 still restrict the usage of these models. Thus, few-shot fine-tuning BERT-style PTMs is more practical for online applications (Gao et al., 2021). iii) Last but not least, although large-scale PTMs have become an important part of the NLP learning pipeline, the slow training and inference speed seriously affects online applications that require higher QPS (Query Per Second) with limited computational resources.

To address these issues, we develop EasyNLP, an NLP toolkit that is designed to make the applications of large PTMs to industrial scenarios more efficiently and effectively. EasyNLP provides knowledge-enhanced pre-training functionalities to improve the knowledge understanding abilities of PTMs. Specifically, it integrates our DKPLM framework (Zhang et al., 2022) that enables the decomposition of knowledge-enhanced pre-training and task-specific learning. Hence, the resulting models can be tuned and deployed in the same way as BERT (Devlin et al., 2019). EasyNLP is equipped with a variety of popular prompt-based few-shot learning algorithms such as PET (Schick and Schütze, 2021) and P-Tuning (Liu et al., 2021b). Particularly, we propose a new few-shot learning paradigm named Contrastive Prompt Tuning (CP-Tuning) (Xu et al., 2022) that eases the manual labor of verbalizer construction based on contrastive learning. Finally, EasyNLP supports several knowledge distillation algorithms that compress large PTMs into small and efficient ones. Among them, the MetaKD algorithm (Pan et al., 2021) can significantly improve the effectiveness of the learned models with cross-domain datasets, which is particular common in industry.

Overall, our EasyNLP toolkit can provide users with large-scale and robust learning functionalities, and is seamlessly connected to the Platform of AI (PAI)¹ products. To demonstrate the useful

¹https://www.alibabacloud.com/product/
of EasyNLP, we also present the results of standard benchmarks and some real-world industrial scenarios to show how EasyNLP brings substantial improvements to these applications.

In a nutshell, the main features of the EasyNLP toolkit include the following aspects:

• Easy-to-use and highly customizable. In addition to providing easy-to-use commands to call cutting-edge NLP models, EasyNLP abstracts customized modules such as AppZoo and ModelZoo to make it easy to build NLP applications. It also features DataHub that provides users with a simple interface to load and process various types of NLP datasets.

• Compatible with open-source community. EasyNLP has rich APIs to support the training of models from other open-source libraries such as Huggingface/Transformers\(^2\) with the PAI’s distributed learning framework. It is also compatible with the PTMs in EasyTransfer ModelZoo\(^3\) (Qiu et al., 2021).

• Product-ready support. EasyNLP is seamlessly integrated to PAI products on Alibaba Cloud to provide full model training and serving experience, including PAI-DSW for model development, PAI-DLC for cloud-native training, PAI-EAS for online serving, and PAI-Designer for zero-code model training.

• Pre-training knowledge-enhanced PTMs. EasyNLP also is equipped with knowledge-enhanced PTMs of various domains. Its pre-training APIs enable users to obtain customized PTMs using their own knowledge bases with just a few lines of codes.

• Deploying large-scale PTMs. EasyNLP provides few-shot learning capabilities based on prompts, allowing users to fine-tune large-scale PTMs with only a few training samples to achieve good results. Meanwhile, it provides knowledge distillation functionalities to help quickly distill large models to small and efficient models for online deployment.

2 Related Work

In this section, we summarize the related work on PTMs, prompt learning and knowledge distillation.

Pre-trained Language Models. PTMs have achieved significant improvements on various tasks by self-supervised pre-training (Qiu et al., 2020). To name a few, BERT (Devlin et al., 2019) learns bidirectional contextual representations by transformer encoders. Other transformer encoder-based PTMs include Transformer-XL (Dai et al., 2019), XLNet (Yang et al., 2019) and many others. The encoder-decoder and auto-regressive decoder architectures are used in T5 (Raffel et al., 2020) and GPT-3 (Brown et al., 2020). Knowledge-enhanced PTMs (Zhang et al., 2019; Liu et al., 2020; Sun et al., 2020) improve language understanding abilities of PTMs via injecting relational triples extracted from knowledge bases.

Prompt Learning for PTMs. Prompt learning models the probability of texts directly as the model prediction results based on language models (Liu et al., 2021a). In the literature, PET (Schick and Schütze, 2021) models NLP tasks as cloze problems and maps the results of the masked language tokens to class labels. Gao et al. (2021) generates discrete prompts from T5 (Raffel et al., 2020) to support prompt discovery. P-Tuning (Liu et al., 2021b) learns continuous prompt embeddings with differentiable parameters. Our CP-Tuning (Xu et al., 2022) optimizes the output results based on contrastive learning, without defining mappings from outputs to class labels.

Knowledge Distillation. Knowledge distillation aims at learning a smaller model from an ensemble or a larger model (Hinton et al., 2015). For large-scale PTMs, DistillBERT (Sanh et al., 2019) and PKD (Sun et al., 2019) applies the distillation loss in the pre-training and fine-tuning stages, separately. TinyBERT (Jiao et al., 2020a) further distills BERT in both stages, considering various types of signals. Due to space limitation, we do not further elaborate other approaches. Our MetaKD method (Pan et al., 2021) further improves the accuracy of the student models by exploiting cross-domain transferable knowledge, which is fully supported by EasyNLP.

3 The EasyNLP Toolkit

In this section, we introduce various aspects of our EasyNLP toolkit in detail.

3.1 Overview

We begin with an overview of EasyNLP in Figure 1. EasyNLP is built upon PyTorch and supports rich
data readers to process data from multiple sources. Users can load any PTMs from ModelZoo and datasets from DataHub, build their applications from AppZoo, or explore its advanced functionalities such as knowledge-enhanced pre-training, knowledge distillation and few-shot learning. The codes can run either in local environments or PAI’s products on the cloud. Users can also explore various solutions on our platform to support real-world applications. In addition, all EasyNLP’s APIs are also released to make it easy for users to customize any kinds of NLP applications.

### 3.2 DataHub, ModelZoo and AppZoo

**DataHub.** DataHub provides users with an interface to load and process various kinds of data. It is compatible with Huggingface datasets\(^4\) as a built-in library that supports unified interface calls and contains datasets of a variety of tasks. Some examples are listed in Table 1. Users can load the required data by specifying the dataset name through the `load_dataset` interface, and then use the `GeneralDataset` interface to automatically process the data into model input. An example of loading and pre-processing the TNEWS dataset, together with its subsequent steps, is shown in Code 1. For user-defined datasets, it is also straightforward to inherit the `GeneralDataset` class to customize the data format.

**ModelZoo.** PTMs such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2020) greatly improve the performance of NLP tasks. To facilitate user deployment of models, ModelZoo provides general pre-trained models as well as our own models for users to use, such as DKPLM (Zhang et al., 2022) of various domains. A few widely-used non-PTM models are also supported, such as Text-CNN (Kim, 2014).

**AppZoo.** To help users build NLP applications more easily with our framework, we further provide a comprehensive NLP application tool named AppZoo. It supports running applications with a few command-line arguments and provides a vari-

\(^4\)https://github.com/huggingface/datasets

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Example of Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence Classification</td>
<td>TNEWS(^1), SogouCA(^6), THUCNews(^3), SogouCS(^8)</td>
</tr>
<tr>
<td>Text Generation</td>
<td>BUSTM(^9), CHID(^10)</td>
</tr>
<tr>
<td>Few-shot / Zero-shot Learning</td>
<td>OntoNotes(^11), Chinese-Literature-NER-RE-Dataset(^12)</td>
</tr>
<tr>
<td>Knowledge-based NLU</td>
<td></td>
</tr>
</tbody>
</table>

```python
from easynlp.dataset import load_dataset, GeneralDataset

dataset = load_dataset('clue', 'tnews')['train']
encoded = GeneralDataset(dataset, 'chinese-bert-base')
model = SequenceClassification('chinese-bert-base')
trainer = Trainer(model, encoded)
trainer.train()
```

Code 1: Load the TNEWS training set and build a text classification application using EasyNLP.

---

\(^1\)https://github.com/CLUEbenchmark/CLUE
\(^2\)http://www.sogou.com/labs/resource/ca.php
\(^3\)http://thuctc.thunlp.org/
\(^4\)https://github.com/CLUEbenchmark/CLUE
\(^5\)https://catalog.ldc.upenn.edu/LDC2013T19
\(^6\)https://www.sogou.com/labs/resource/cs.php
\(^7\)https://github.com/xiaobu-coai/BUSTM
\(^8\)https://github.com/chujiezheng/ChID-Dataset
\(^9\)https://github.com/CLUEbenchmark/CLUE
\(^10\)https://catalog.ldc.upenn.edu/LDC2013T19
\(^11\)https://github.com/lancopku/Chinese-Literature-NER-RE-Dataset

---
Code 2: AppZoo for training a BERT-based text classifier using EasyNLP.

3.3 In-house Developed Algorithms

In this section, we introduce in-house developed algorithms in EasyNLP. All these algorithms have been tested in real-world applications.

3.3.1 Knowledge-enhanced Pre-training

Knowledge-enhanced pre-training improves the performance of PTMs by injecting the relational facts from knowledge bases. Yet, a lot of existing works require additional knowledge encoders during pre-training, fine-tuning and inference (Zhang et al., 2019; Liu et al., 2020; Sun et al., 2020).

The proposed DKPLM paradigm (Zhang et al., 2022) decomposes the knowledge injection process. For DKPLM, knowledge injection is only applied during pre-training, without introducing extra parameters as knowledge encoders, alleviating the significant computational burdens for users. Meanwhile, during fine-tuning and inference stages, our model can be utilized in the same way as that of BERT (Devlin et al., 2019) and other plain PTMs, which facilitates the model fine-tuning and deployment in EasyNLP and other environments.

Specifically, the DKPLM framework introduces three novel techniques for knowledge-enhanced pre-training. It recognizes long-tail entities from text corpora for knowledge injection only, avoiding learning too much redundant and irrelevant information from knowledge bases (Zhang et al., 2021). Next, the representations of entities are replaced by “pseudo token representations” derived from knowledge bases, without introducing any extra parameters to DKPLM. Finally, a relational knowledge decoding task is introduced to force the model to understand what knowledge is injected.

In EasyNLP, we provide the entire pre-training pipeline of DKPLM for users. In addition, a collection of pre-trained DKPLMs for specific domains have been registered in ModelZoo for supporting domain-specific applications.

3.3.2 Few-shot Learning for PTMs

For low-resource scenarios, prompt-based learning leverages prompts as task guidance for effective few-shot fine-tuning. In EasyNLP, to facilitate easy few-shot learning, we integrate PET (Schick and Schütze, 2021) and P-Tuning (Liu et al., 2021b) into AppZoo that allow users call the algorithms in the similar way compared to standard fine-tuning.

It should be further noted that either PET or P-Tuning require the explicit handcraft of verbalizers, which is a tedious process and may lead to unstable results. Our CP-Tuning approach (Xu et al., 2022) enables few-shot fine-tuning PTMs without the manual engineering of task-specific prompts and verbalizers. A pair-wise cost-sensitive contrastive learning is introduced to achieve verbalizer-free class mapping by learning to distinguish different classes. Users can also explore CP-Tuning in AppZoo for any tasks that classical prompt-based methods support.

3.3.3 Knowledge Distillation for PTMs

The large model size and the long inference time hinder the deployment of large-scale PTMs to resource-constrained applications. In EasyNLP, we provide a complete learning pipeline for knowledge distillation, including data augmentation for training sets, logits extraction from teacher models and distilled training of student models.

In addition, we notice that a majority of existing approaches focus on a single domain only. The proposed MetaKD algorithm (Pan et al., 2021) explicitly leverages the cross-domain transferable knowledge to improve the accuracy of student models. It first obtain a meta-teacher model to capture transferable knowledge at both instance-level and feature-level from multiple domains. Next, a meta-distillation algorithm is employed to learn single-domain student models with selective signals from the meta-teacher. In EasyNLP, the MetaKD process is implemented as a general feature for any types of BERT-style PTMs.
4 System Evaluations and Applications

In this section, we empirically examine the effectiveness and efficiency of the EasyNLP toolkit on both public datasets and industrial applications.

4.1 CLUE and GLUE Benchmarks

In order to validate the effectiveness of EasyNLP on model fine-tuning, we fine-tune PTMs on the CLUE and GLUE benchmarks (Wang et al., 2019; Xu et al., 2020). For all tasks, we use a limited hyper-parameter search space, with batch sizes in \{8, 16, 32, 48\}, sequence length in \{128, 256\} and learning rates in \{1e−5, 2e−5, 3e−5, 4e−5, 5e−5\}. The underlying PTMs include BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). We also evaluate MacBERT (Cui et al., 2020) for the Chinese benchmark CLUE. We report the results over the development sets of each task in the two benchmarks, shown in Tables 2 and 3, respectively. The obtained comparable performance has shown the reliability of EasyNLP, which achieves similar performance compared to other open-source frameworks and their original implementations.

4.2 Evaluation of Knowledge-enhanced Pre-training

We report the performance of DKPLM over zero-shot knowledge probing tasks, including LAMA (Petroni et al., 2019) and LAMA-UHN (Pörner et al., 2019), with the results summarized in Table 4. Compared to strong baselines (i.e., COLAKE (Sun et al., 2020) K-Adapter (Wang et al., 2021a) and KEPLER (Wang et al., 2021b)), we see that DKPLM achieves state-of-the-art results over three datasets (+1.57% on average). The result of DKPLM is only 0.1% lower than K-Adapter, without using any T-REx training data and larger backbones. We can see that our pre-training process based on DKPLM can effectively store and understand factual relations from knowledge bases.

Industrial Applications. Based on the proposed DKPLM framework (Zhang et al., 2022), we have pre-trained a series of domain-specific PTMs to provide model service inside Alibaba Group, such as medical and finance domains, and observed consistent improvement in downstream NLP tasks. For example, the medical-domain DKPLM improves the accuracy of a medical named entity recognition task by over 3%, compared to the standard BERT model (Devlin et al., 2019). The pre-trained model (named pai-dkplm-medical-base-zh) has also been released in our EasyNLP ModelZoo.

4.3 Evaluations of Few-shot Learning

We compare CP-Tuning (Xu et al., 2022) against several prompt-based fine-tuning approaches including PET (Schick and Schütze, 2021), LM-BFF (Gao et al., 2021) (in three settings where “Auto T”, “Auto L” and “Auto T+L” refer to the prompt-tuned PTM with automatically generated templates, label words and both, respectively) and P-Tuning (Liu et al., 2021b). The experiments are conducted over several text classification datasets in a 16-shot learning setting. The underlying PTM is RoBERTa (Liu et al., 2019). Readers can refer to Xu et al. (2022) for more details. From the results in Table 5, we can see that the performance gains of CP-Tuning over all the tasks are consistent, compared to state-of-the-art methods.

Industrial Applications. For business customer...
Table 4: The performance on LAMA knowledge probing datasets. Note that K-Adapter is trained based on a large-scale model and uses a subset of T-REx as its training data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ELMo</th>
<th>BERT</th>
<th>RoBERTa</th>
<th>ColLAKE</th>
<th>K-Adapter*</th>
<th>KEPLER</th>
<th>DKPLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google-RE</td>
<td>2.2%</td>
<td>11.4%</td>
<td>5.3%</td>
<td>9.5%</td>
<td>7.0%</td>
<td>7.3%</td>
<td>10.8%</td>
</tr>
<tr>
<td>UHN-Google-RE</td>
<td>2.3%</td>
<td>5.7%</td>
<td>2.2%</td>
<td>4.9%</td>
<td>3.7%</td>
<td>4.1%</td>
<td>5.4%</td>
</tr>
<tr>
<td>T-REx</td>
<td>0.2%</td>
<td>32.5%</td>
<td>24.7%</td>
<td>28.8%</td>
<td>29.1%</td>
<td>24.6%</td>
<td>32.0%</td>
</tr>
<tr>
<td>UHN-T-REx</td>
<td>0.2%</td>
<td>23.3%</td>
<td>17.0%</td>
<td>20.4%</td>
<td>23.0%</td>
<td>17.1%</td>
<td>22.9%</td>
</tr>
</tbody>
</table>

Table 5: Comparison between CP-Tuning and baselines over the testing sets in terms of accuracy (%).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Fine-tuning</td>
<td>78.62</td>
<td>76.17</td>
<td>72.48</td>
<td>64.40</td>
<td>63.01</td>
<td>62.32</td>
<td>52.28</td>
<td>86.82</td>
<td>69.51</td>
</tr>
<tr>
<td>PET</td>
<td>92.06</td>
<td>87.13</td>
<td>87.13</td>
<td>66.23</td>
<td>70.34</td>
<td>64.38</td>
<td>65.56</td>
<td>91.28</td>
<td>78.01</td>
</tr>
<tr>
<td>LM-BFF (Auto T)</td>
<td>90.60</td>
<td>87.57</td>
<td>90.76</td>
<td>66.72</td>
<td>65.25</td>
<td>68.87</td>
<td>65.99</td>
<td>91.61</td>
<td>78.42</td>
</tr>
<tr>
<td>LM-BFF (Auto L)</td>
<td>90.55</td>
<td>85.51</td>
<td>91.11</td>
<td>67.75</td>
<td>70.92</td>
<td>66.22</td>
<td>66.35</td>
<td>90.48</td>
<td>78.61</td>
</tr>
<tr>
<td>LM-BFF (Auto T+L)</td>
<td>91.42</td>
<td>86.84</td>
<td>90.40</td>
<td>66.81</td>
<td>61.61</td>
<td>61.89</td>
<td>66.79</td>
<td>90.72</td>
<td>77.06</td>
</tr>
<tr>
<td>P-tuning</td>
<td>91.42</td>
<td>87.41</td>
<td>90.90</td>
<td>71.23</td>
<td>66.77</td>
<td>63.42</td>
<td>67.15</td>
<td>89.10</td>
<td>78.43</td>
</tr>
<tr>
<td>CP-Tuning</td>
<td>93.35</td>
<td>89.43</td>
<td>91.57</td>
<td>72.60</td>
<td>73.56</td>
<td>69.22</td>
<td>67.22</td>
<td>92.27</td>
<td>81.24</td>
</tr>
</tbody>
</table>

Table 6: Evaluation of MetaKD over Amazon reviews and MNLI in terms of averaged accuracy (%).

<table>
<thead>
<tr>
<th>Method</th>
<th>Amazon</th>
<th>MNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-s</td>
<td>87.9</td>
<td>81.9</td>
</tr>
<tr>
<td>BERT-mix</td>
<td>89.5</td>
<td>84.4</td>
</tr>
<tr>
<td>BERT-mtl</td>
<td>89.8</td>
<td>84.2</td>
</tr>
<tr>
<td>BERT-s → TinyBERT</td>
<td>86.7</td>
<td>79.3</td>
</tr>
<tr>
<td>BERT-mix → TinyBERT</td>
<td>87.3</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT-mtl → TinyBERT</td>
<td>87.7</td>
<td>79.7</td>
</tr>
<tr>
<td>MetaKD</td>
<td>89.4</td>
<td>80.4</td>
</tr>
</tbody>
</table>

4.4 Evaluations of Knowledge Distillation

We further report the performance of MetaKD (Pan et al., 2021) on Amazon reviews (Blitzer et al., 2007) and MNLI (Williams et al., 2018), where the two datasets contain four and five domain instances, respectively. In the experiments, we train the meta-teacher over multi-domain training sets, and distill the meta-teacher to each of all the domains. The teacher model is BERT-base (with 110M parameters), while the student model is BERT-tiny (with 14.5M parameters). Table 6 shows the performance of baselines and MetaKD, in terms of averaged accuracy across domains. BERT-s refers to a single BERT teacher trained on each domain. BERT-mix is one BERT teacher trained on the mixture of all domain data. BERT-mtl is one teacher trained by multi-task learning over all domains. For distillation, “→ TinyBERT” means using the KD method described in Jiao et al. (2020b) to distill the corresponding teacher model. The results show that MetaKD significantly reduces the model size while preserving a similar performance. For more details, we refer the readers to Pan et al. (2021).

Industrial Applications. Distilled PTMs have been widely used inside Alibaba Group due to the high QPS requirements of online e-commerce applications. For example, in the AliMe chatbot (Qiu et al., 2017), we distill the BERT-based query intent detection model from the base version to the tiny version, resulting in 7.2x inference speedup while the accuracy is only decreased by 1%.

5 Conclusion

In this paper, we introduced EasyNLP, a toolkit that is designed to make it easy to develop and deploy deep NLP applications based on PTMs. It supports a comprehensive suite of NLP algorithms and features knowledge-enhanced pre-training, knowledge distillation and few-shot learning functionalities for large-scale PTMs. Currently, EasyNLP has powered a number of business units inside Alibaba Cloud and provided NLP service on the cloud. The toolkit has been open-sourced to promote research and development for NLP applications.
**Broader Impact**

EasyNLP is a comprehensive toolkit for building various NLP applications to support industrial scenarios. It has been seamlessly integrated into the PAI products, and has been released to the open-source community. EasyNLP is also highly beneficial for academia, as it integrates state-of-the-art methods and models to make it easy for researchers to benchmark and develop their own algorithms.

**Acknowledgments**

The authors would like to thank Haojie Pan, Peng Li, Boyu Hou, Xiaoming Chen, Xiaodan Wang, Xiangru Zhu and many other members of the Alibaba PAI team for their contribution and suggestions on building the EasyNLP toolkit. This work is also partially supported by Alibaba Group through Alibaba Innovative Research Program and Alibaba Research Intern Program.

**References**

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In ACL.


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


Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Ye Chen, Xue Hu, Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanying Li, Yina Patterson, Zouyu Tian, Yiheng Zhang, He Zhou, Shaoxuehua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zheng Liang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In COLING, pages 4762–4772.


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

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In ACL, pages 1441–1451.