LERT: A Linguistically-motivated Pre-trained Language Model

Yiming Cui[1,2,†], **Wanxiang Che**[1], **Shijin Wang**[2,3], **Ting Liu**[1]
[1]Research Center for SCIR, Harbin Institute of Technology, Harbin, China
[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, Beijing, China
[3]iFLYTEK AI Research (Central China), Wuhan, China
[†]ymcui@ieee.org

Abstract

Pre-trained Language Model (PLM) has become a representative foundation model in the natural language processing field. Most PLMs are trained with linguistic-agnostic pre-training tasks on the surface form of the text, such as the masked language model (MLM). To further empower the PLMs with richer linguistic features, in this paper, we aim to propose a simple but effective way to learn linguistic features for pre-trained language models. We propose LERT, a pre-trained language model that is trained on three types of linguistic features along with the original MLM pre-training task, using a linguistically-informed pre-training (LIP) strategy. We carried out extensive experiments on ten Chinese NLU tasks, and the experimental results show that LERT could bring significant improvements over various comparable baselines. Furthermore, we also conduct analytical experiments in various linguistic aspects, and the results prove that the design of LERT is valid and effective.[1]

1 Introduction

Pre-trained Language Model (PLM) has been proven to be a successful way for text representation, which considers rich contextual information. Among several types of pre-trained language models, auto-encoding PLMs, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b), are relatively popular for natural language understanding (NLU) tasks. Unlike the auto-regressive PLMs (e.g., GPT (Radford et al., 2018)) that use a standard language model as the training objective, auto-encoding PLMs largely rely on pre-training tasks to learn contextual information. Masked language model (MLM), which was first proposed in BERT, has been a dominant pre-training task for auto-encoding PLMs, such as RoBERTa, ALBERT (Lan et al., 2020), ERNIE (Sun et al., 2019), DeBERTa (He et al., 2021), etc., demonstrating its broad generalizability in learning text representations. The MLM task learns to recover word information from the masked text, where the masked word is usually chosen randomly, indicating that MLM is a linguistic-agnostic pre-training task without explicit utilization of linguistic knowledge.

Though it is widely perceived that the pre-trained language model entails rich linguistic knowledge (Jawahar et al., 2019), some researchers propose to further include external knowledge in PLMs. Specifically, to incorporate linguistic knowledge into the pre-trained language model, various efforts have been made in the community, such as incorporating structural knowledge (Zhou et al., 2020; Xu et al., 2021), including additional linguistic tasks (Zhang et al., 2021), etc. Though various efforts have been made, the previous work has several limitations. Most of these works only focus on including several linguistic features in PLM without carefully analyzing how individual features contribute to the overall performance and the relations between different tasks. Also, the implementations are relatively complex, as structural knowledge cannot be directly applied into PLMs.

To alleviate the issues above, in this paper, we leverage the traditional natural language processing method to explicitly include more linguistic knowledge, creating weakly-supervised data for model pre-training. Also, to investigate whether pre-trained language models can benefit from explicitly injecting linguistic knowledge, in this paper, we propose a new pre-trained language model called **LERT** (**L**inguistically-motivated bidirectional **E**ncoder **R**epresentation from **T**ransformer). LERT is trained on the masked language model as well as three types of linguistic tasks, including part-of-speech (POS) tagging, named entity recognition (NER), and dependency parsing (DEP), forming a multi-task pre-training scheme. Further-

[1]Pre-print version, subjected to changes. Resources are available at https://github.com/ymcui/LERT

more, to balance the learning speed for each pre-training task, we propose a linguistically-informed pre-training (LIP) strategy, which learns fundamental linguistic knowledge faster than the high-level ones. With thorough ablations and analyses, LERT has proven effective on various natural language understanding tasks over comparable baselines. The contributions of this paper are listed as follows.

- We propose a simple way to incorporate three types of linguistic features for pre-trained language models with a linguistically-informed pre-training (LIP) strategy.
- With extensive and robust experiments on ten popular Chinese natural language understanding tasks, LERT yields significant improvements over comparable baselines. Several analyses also prove the effectiveness of LERT.
- The resources are made publicly available to further facilitate our research community.

2 Related Work

The recent advancement of natural language processing largely owes to the development of text representations. Speech signals can be represented by waves, and images can be represented by pixels, where they all have clear physical concepts and can be directly represented in computers. However, when it comes to natural language, it has no exact representation for a specific semantic. Thus, a major research topic in NLP is to find a better way for text representation. In the last decade, static word embedding has been a dominant text representation method in NLP, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). However, these representations cannot deal with the problem of polysemy. Later, ELMo (Peters et al., 2018) was proposed to solve this issue, which models the text in recurrent neural networks, and word representation can be adjusted by its context. Transformer-based (Vaswani et al., 2017) neural networks have proven effective among various NLP tasks. A combination of the transformer model and text representation has led to the recent emergence of pre-trained language models. Pre-trained language model (PLM), such as BERT and GPT, uses deep transformer models to encode the text in a contextual way, which can be applied to a wide range of natural language processing tasks. The training of PLM only requires large-scale unlabeled text with self-supervised tasks, such as the masked language model. Though linguistic knowledge is not explicitly injected, various PLMs achieve significant improvements on many NLP tasks.

One of the main reasons that make PLMs successful is that pre-trained language models learn better text semantics and entail linguistic knowledge, though they are not explicitly learned in the self-supervised task, which is commonly perceived by the community. For example, Jawahar et al. (2019) discovered that the intermediate layers of BERT capture rich linguistic information. Kovaleva et al. (2019) focuses on the multi-head self-attention mechanism itself to demonstrate its redundancies. Liu et al. (2019a) investigated the transferability of contextual representations with several linguistic probing tasks. Hewitt and Manning (2019) propose a structural probe for finding syntax information in pre-trained language models. These works have brought us a better understanding of which types of linguistic features are learned in PLMs.

Some researchers also tried incorporating linguistic features in pre-trained language models to further improve their performance on downstream tasks. Zhou et al. (2020) propose LIMIT-BERT, which incorporates five linguistic tasks: part-of-speech, constituent and dependency parsing, span, and dependency semantic role labeling (SRL). Xu et al. (2021) propose a syntax-enhanced pre-trained model, which incorporates a syntax-aware attention layer during both the pre-training and fine-tuning stages. Zhang et al. (2021) utilizes part-of-speech tagging and named entity recognition as additional linguistic tasks during pre-training. Liu et al. (2021) propose LEBERT for Chinese sequence labeling, which incorporates external knowledge into BERT layers. Zhang et al. (2022) propose CK-BERT, which uses linguistic-aware MLM and contrastive multi-hop relation model for pre-training.

Unlike previous works that either depend on incorporating structural knowledge with complex model design or without a clear exposition of the contribution by each linguistic feature, in this paper, we proposed LERT, which aims to directly utilize the linguistic tags for multi-task pre-training. The pre-training task is chosen by careful analysis, and the LERT also benefits from a linguistically-informed pre-training scheme, which is in line with intuitive thinking. The experiments and analyses present a clear contribution of each linguistic feature as well as other components to explicitly allow

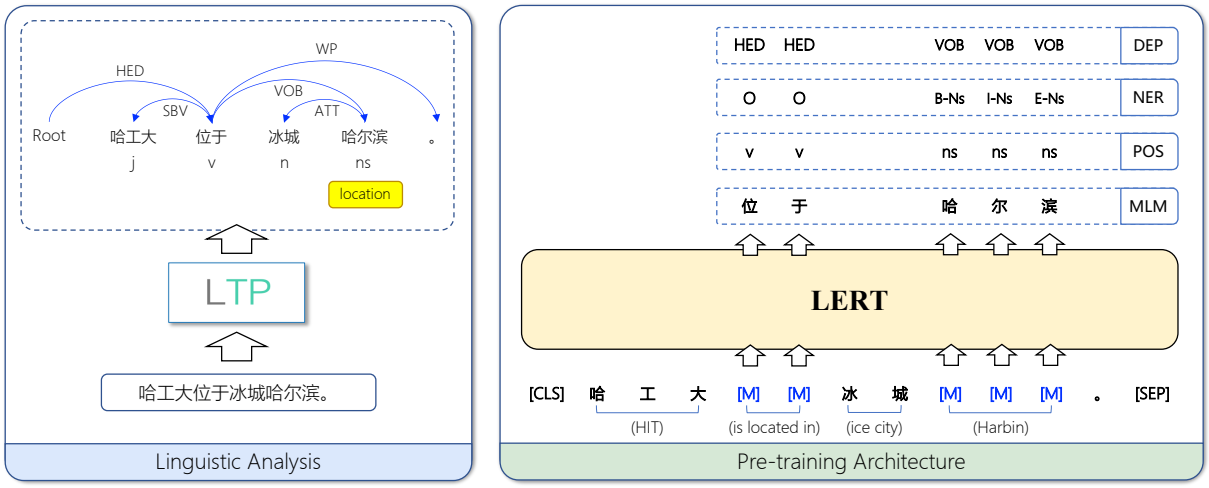

Figure 1: Overview of LERT. We only use one text segment in the input for simplicity, i.e., we still use two text segments for implementation, separated by [SEP] token.

us to understand which type of linguistic feature is the most helpful in creating a better pre-trained language model.

3 LERT

3.1 Overview

An overview of the proposed LERT is depicted in Figure 1. The formulation of LERT is simple and straightforward. Firstly, we perform linguistic analysis on the given input text to get word segmentation information and extract its linguistic features. The word segmentation information is used to perform Chinese whole word masking (wwm) and N-gram masking (Cui et al., 2021) (identical to MacBERT (Cui et al., 2021), PERT (Cui et al., 2022), etc.) in masked language model task. The linguistic features are used for linguistic task pre-training. Then we use the extracted linguistic features to perform multi-task pre-training along with the original MLM task, which linguistically-informed pre-training scheme.

3.2 Linguistic Features

In this paper, we aim to utilize linguistic features in a simple way. To meet this criterion, the generated linguistic feature should have two characteristics: high-accuracy and uniqueness. High-accuracy means that the linguistic feature should be highly reliable in terms of tagging performance. Though current language analysis tools, such as LTP (Che et al., 2010), Stanford CoreNLP (Manning et al., 2014), etc., are capable of analyzing the linguistic features for the text, not all of the features are highly accurate. Uniqueness means that each input token should have exactly one target tag for a specific linguistic feature, whereas most tree-based or graph-based linguistic analyses do not meet this standard and requires further complex processing.

Considering both conditions, in this paper, we use LTP (Che et al., 2010) for annotating linguistic tags for the input text with three types of linguistic features, i.e., part-of-speech (POS), named entity recognition (NER), and dependency parsing (DEP). These three types of linguistic features are relatively fundamental and achieve good performance on tagging and meet one-to-one tagging conditions.[2] A complete list of linguistic tags are depicted in Table 1. Specifically,

- **POS**: Each input token is assigned to a unique POS tag, resulting in 28 types.
- **NER**: We use the "BIEOS" tagging scheme to annotate input tokens with NER information, resulting in 13 types.
- **DEP**: We perform syntactic dependency parsing on the input sequence. Note that we attribute the relation label to its dependent (not head) to ensure each token has a unique label, resulting in 14 types.

After getting these linguistic labels for each input token, we can treat them as weakly-supervised labels for pre-training, where we illustrate pre-training tasks in the next.

[2]POS: 98.4% (P), NER: 91.7% (F), DEP: 84.8 (UAS), according to http://ltp.ai/docs/ltp3.x/theory.html

Type (#)	Tags (abbreviation)
POS (28)	noun (n), verb (v), punctuation (wp), auxiliary (u), adverb (d), adjective (a), number (m), preposition (p), pronoun (r), geographical name (ns), conjunction (c), quantity (q), temporal noun (nt), person name (nh), direction noun (nd), abbreviation (j), idiom (i), other noun-modifier (b), organization name (ni), other proper noun (nz), location noun (nl), descriptive words (z), suffix (k), foreign words (ws), onomatopoeia (o), prefix (h), exclamation (e), non-lexeme (x)
NER (13)	outside (O), single (S-Ni/Ns/Nh), organization names (B/I/E-Ni), person names (B/I/E-Nh), location names (B/I/E-Ns)
DEP (14)	attribute (ATT), punctuation (WP), adverbial (ADV), verb-object (VOB), subject-verb (SBV), coordinate (COO), right adjunct (RAD), head (HED), preposition-object (POB), complement (CMP), left adjunct (LAD), fronting-object (FOB), double (DBL), indirect-object (IOB)

Table 1: A list of linguistic tags used in LERT.

3.3 Model Pre-training

LERT is trained on the masked language model as well as three linguistic tasks, forming a multi-task training scheme.

3.3.1 MLM Task

For the MLM task, we follow most of the previous works that only make predictions on the masked positions[3] instead of the whole input sequence. We denote the last hidden layer representation of L-layer transformer as $\boldsymbol{H} \in \mathbb{R}^{N \times d}$ (N is the length of input sequence, d is hidden size), and a subset of representations w.r.t. masked positions as $\boldsymbol{H}^m \in \mathbb{R}^{k \times d}$ (k is the number of masked positions), where $\boldsymbol{H}^m \subset \boldsymbol{H}$. Then we use a fully-connected layer, followed by a layer normalization layer on $\boldsymbol{H}^m$.

$$\tilde{\boldsymbol{H}}^m = \mathbf{LayerNorm}(\mathbf{FFN}(\boldsymbol{H}^m)) \quad (1)$$

We use the input word embedding matrix $\boldsymbol{E} \in \mathbb{R}^{V \times d}$ (V is the vocabulary size) to project the $\tilde{\boldsymbol{H}}^m$ into vocabulary space, and use the softmax function to get normalized probabilities.

$$\boldsymbol{p}_i = \mathbf{softmax}(\tilde{\boldsymbol{H}}_i^{\mathrm{m}} \boldsymbol{E}^\top + \boldsymbol{b}), \;\; \boldsymbol{p}_i \in \mathbb{R}^V \quad (2)$$

Finally, we use the standard cross-entropy loss to optimize the MLM pre-training task.

$$\mathcal{L}_{\mathrm{MLM}} = -\frac{1}{M} \sum_{i=1}^{M} \boldsymbol{y}_i \log \boldsymbol{p}_i \quad (3)$$

3.3.2 Linguistic Tasks

For each linguistic task, we treat it as a classification task. Each input token is projected to its linguistic feature (POS, NER, and DEP), which was annotated using the method described in the previous section. Specifically, given the representation $\tilde{\boldsymbol{H}}^m$, we use a fully-connected layer to project it into linguistic labels for each task.

$$\boldsymbol{p}_i^\star = \mathbf{softmax}(\tilde{\boldsymbol{H}}_i^{\mathrm{m}} \boldsymbol{W}^{\star\top} + \boldsymbol{b}^\star), \;\; \boldsymbol{p}_i \in \mathbb{R}^{V^\star} \quad (4)$$

In Equation 4, the $\star$ can be one of three linguistic tasks, and $V^\star$ denotes the number of linguistic labels for each task. We use standard cross-entropy loss to optimize each linguistic task.

3.3.3 Linguistically-informed Pre-training

Finally, the overall training loss is formulated as follows, where $\lambda_i \in [0, 1]$ is the scaling factor to the respective loss $\mathcal{L}_i$ for each linguistic task (POS, NER, and DEP).

$$\mathcal{L} = \mathcal{L}_{\mathrm{MLM}} + \sum_i \lambda_i \mathcal{L}_i, \;\; i \in \{\mathrm{P}, \mathrm{N}, \mathrm{D}\} \quad (5)$$

A vanilla pre-training scheme is to treat all subtasks as equal, resulting in the following equation.

$$\mathcal{L} = \mathcal{L}_{\mathrm{MLM}} + \mathcal{L}_{\mathrm{P}} + \mathcal{L}_{\mathrm{N}} + \mathcal{L}_{\mathrm{D}} \quad (6)$$

Intuitively, the MLM task is the most important one among all subtasks. However, how do we decide the scaling factor λ for each linguistic task?

In this paper, we propose a linguistically-informed pre-training (LIP) strategy to tackle this issue. By looking into these linguistic features, they are not completely equivalent. The NER feature depends on the output of POS tagging, while the DEP feature depends on both POS and NER tagging. We conjecture that POS is the most fundamental linguistic feature, followed by NER and DEP. In light of their dependencies, we assign different learning speeds for each linguistic feature, yielding faster learning of POS than NER and DEP. This is similar to human learning, where we usually learn basic things first and then the dependent high-level knowledge.

Formally, the loss scaling parameters are determined by the current training step t and constant

[3]Note that "masked positions" includes three types of masking: "replace with [MASK]", "keep original word", and "replace with the random word".

end step for scaling $T_\star$ that control the learning speed for each linguistic task.

$$\lambda_\star = \min\{\frac{t}{T_\star}, 1\}, \star \in \{\mathrm{P}, \mathrm{N}, \mathrm{D}\} \quad (7)$$

Specifically, in this paper, we set $T_\star$ as 1/6, 1/3, and 1/2 of the total training steps for POS, NER, and DEP features, respectively. After 1/2 of the total training steps, the training loss will become Equation 6, where all tasks contribute equally to the overall loss. In this way, the POS features learn the most quickly, followed by NER and DEP. We empirically find this strategy yields better performance, and detailed analysis also proves our strategy is effective (Section 5.2) and in line with intuitive thoughts.

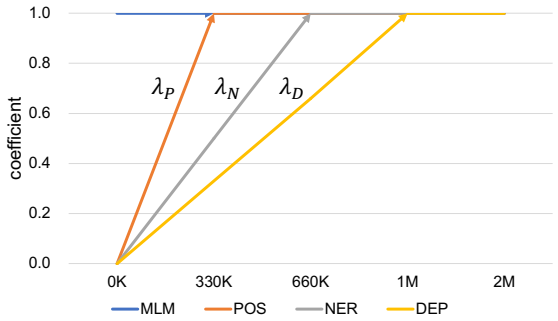

Figure 2: Linguistically-informed pre-training strategy for LERT (with a total pre-training step of 2M).

4 Experiments

4.1 Setups for Pre-training

In this paper, we mainly train three LERT models. The basic information is listed in Tabel 2. Other pre-training setups are illustrated as follows.

- **Data**: We use the training data as in MacBERT and PERT. It consists of the Chinese Wikipedia dump, encyclopedia, community question answering, news articles, etc., resulting in 5.4B words and taking about 20G of disk space.
- **Text processing**: We use WordPiece tokenizer (Wu et al., 2016) as in BERT and similar variants. All linguistic processing (such as word segmentation, tagging, etc.) is done with LTP (Che et al., 2010). We directly use the same vocabulary as in Chinese BERT-base with 21,128 entries.
- **Optimization**: We use ADAM (Kingma and Ba, 2014) with weight decay (rate = 0.1) optimizer using an initial learning rate of 1e-4. Each model is trained on 2M steps with the first 10K steps of linear warmup for learning rate. All models are trained from scratch.
- **Others**: The maximum sequence length is 512. The overall masking ratio is set as 15%.
- **Training device**: All models are trained on a single Cloud TPU v3-8 (128G HBM) with gradient accumulation (if necessary).

Model	Params	Layers	Hid.	A.H.	Batch
$\text{LERT}_{\text{small}}$	15M	12	256	4	1024
$\text{LERT}_{\text{base}}$	102M	12	768	12	416
$\text{LERT}_{\text{large}}$	325M	24	1024	16	256

Table 2: Model structure for different sizes of LERT. Hid: hidden size, A.H.: attention heads.

4.2 Setups for Fine-tuning Tasks

Following previous works (Cui et al., 2021, 2022), we examine LERT's performance on ten natural language understanding tasks, including machine reading comprehension (MRC), text classification (TC), named entity recognition (NER), etc. Specifically,

- **MRC** (2): CMRC 2018 (Cui et al., 2019), DRCD (Shao et al., 2018).
- **TC** (6): XNLI (Conneau et al., 2018), LCQMC (Liu et al., 2018), BQ Corpus (Chen et al., 2018), ChnSentiCorp (Tan and Zhang, 2008), TNEWS (Xu et al., 2020), OCNLI (Hu et al., 2020).
- **NER** (2): MSRA-NER (SIGHAN 2006) (Levow, 2006), People's Daily (PD)[4].

Dataset	MaxL	Ep.	Train	Dev	Test
CMRC-18	512	3/2/1	10K	3.2K	4.9K
DRCD	512	5/2/3	27K	3.5K	3.5K
XNLI	128	5/2/2	392K	2.5K	5K
LCQMC	128	5/3/3	240K	8.8K	12.5K
BQ Corpus	128	5/3/2	100K	10K	10K
CSC	256	5	9.6K	1.2K	1.2K
TNEWS	128	5	53.3K	10K	10K
OCNLI	128	5/5/3	56K	3K	3K
MSRA	256	10	45K	-	3.4K
PD	256	10	51K	4.6K	-

Table 3: Hyper-parameter settings and data statistics for fine-tuning tasks. MaxL: sequence max length, Ep: training epochs (small/base/large).

[4] https://github.com/ProHiryu/bert-chinese-ner

System	CMRC 2018						DRCD			
	Dev (EM/F1)		Test (EM/F1)		Challenge (EM/F1)		Dev (EM/F1)		Test (EM/F1)	
$\text{BERT}_{\text{base}}$	67.1 (65.6)	85.7 (85.0)	71.4 (70.0)	87.7 (87.0)	24.0 (20.0)	47.3 (44.6)	85.0 (84.5)	91.2 (90.9)	83.6 (83.0)	90.4 (89.9)
$\text{RoBERTa}_{\text{base}}$	67.4 (66.5)	87.2 (86.5)	72.6 (71.4)	89.4 (88.8)	26.2 (24.6)	51.0 (49.1)	86.6 (85.9)	92.5 (92.2)	85.6 (85.2)	92.0 (91.7)
$\text{ELECTRA}_{\text{base}}$	68.4 (68.0)	84.8 (84.6)	73.1 (72.7)	87.1 (86.9)	22.6 (21.7)	45.0 (43.8)	87.5 (87.0)	92.5 (92.3)	86.9 (86.6)	91.8 (91.7)
$\text{MacBERT}_{\text{base}}$	68.5 (67.3)	87.9 (87.1)	73.2 (72.4)	89.5 (89.2)	**30.2** (26.4)	54.0 (52.2)	89.4 (89.2)	94.3 (94.1)	89.5 (88.7)	93.8 (93.5)
$\text{PERT}_{\text{base}}$	68.5 (68.1)	87.2 (87.1)	72.8 (72.5)	89.2 (89.0)	28.7 (**28.2**)	55.4 (53.7)	89.5 (88.9)	93.9 (93.6)	89.0 (88.5)	93.5 (93.2)
$\text{LERT}_{\text{base}}$	**69.2 (68.4)**	**88.1 (87.9)**	**73.5 (72.8)**	**89.7 (89.4)**	27.7 (26.7)	**55.9 (54.6)**	**90.5 (90.2)**	**95.1 (94.9)**	**90.5 (90.2)**	**94.9 (94.7)**
$\text{RoBERTa}_{\text{large}}$	68.5 (67.6)	88.4 (87.9)	74.2 (72.4)	90.6 (90.0)	31.5 (30.1)	60.1 (57.5)	89.6 (89.1)	94.8 (94.4)	89.6 (88.9)	94.5 (94.1)
$\text{ELECTRA}_{\text{large}}$	69.1 (68.2)	85.2 (84.5)	73.9 (72.8)	87.1 (86.6)	23.0 (21.6)	44.2 (43.2)	88.8 (88.7)	93.3 (93.2)	88.8 (88.2)	93.6 (93.2)
$\text{MacBERT}_{\text{large}}$	70.7 (68.6)	88.9 (88.2)	74.8 (73.2)	90.7 (90.1)	31.9 (29.6)	60.2 (57.6)	91.2 (90.8)	95.6 (95.3)	**91.7** (90.9)	95.6 (95.3)
$\text{PERT}_{\text{large}}$	**72.2 (71.0)**	89.4 (88.8)	**76.8 (75.5)**	90.7 (90.4)	**32.3 (30.9)**	59.2 (58.1)	90.9 (90.8)	95.5 (95.2)	91.1 (90.7)	95.2 (95.1)
$\text{LERT}_{\text{large}}$	71.2 (70.5)	**89.5 (89.1)**	75.6 (75.1)	**90.9 (90.6)**	32.3 (29.7)	**61.2 (59.2)**	**91.6 (91.3)**	**96.1 (95.8)**	91.5 (**91.1**)	**95.9 (95.5)**

Table 4: Experimental results on MRC tasks: CMRC 2018 (Simplified Chinese) and DRCD (Traditional Chinese). We report both the maximum and average scores (in parenthesis) for each set. Overall best performances are depicted in boldface (base-level and large-level are marked individually).

System	XNLI		LCQMC		BQ Corpus		ChnSentiCorp		TNEWS	OCNLI
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Dev
$\text{BERT}_{\text{base}}$	79.4 (78.6)	78.7 (78.3)	89.6 (89.2)	87.1 (86.6)	**86.4** (85.5)	**85.3** (84.8)	**95.4** (94.6)	95.3 (94.8)	57.0 (56.6)	76.0 (75.3)
$\text{RoBERTa}_{\text{base}}$	80.0 (79.2)	78.8 (78.3)	89.0 (88.7)	86.4 (86.1)	86.0 (85.4)	85.0 (84.6)	94.9 (94.6)	95.6 (94.9)	57.4 (56.9)	76.5 (76.0)
$\text{ELECTRA}_{\text{base}}$	77.9 (77.0)	78.4 (77.8)	**90.2 (89.8)**	**87.6 (87.3)**	84.8 (84.7)	84.5 (84.0)	93.8 (93.0)	94.5 (93.5)	56.1 (55.7)	76.1 (75.8)
$\text{MacBERT}_{\text{base}}$	**80.3 (79.7)**	79.3 (78.8)	89.5 (89.3)	87.0 (86.5)	86.0 (85.5)	85.2 (**84.9**)	95.2 (**94.8**)	95.6 (94.9)	57.4 (**57.1**)	77.0 (76.5)
$\text{PERT}_{\text{base}}$	78.8 (78.1)	78.1 (77.7)	88.8 (88.3)	86.3 (86.0)	84.9 (84.8)	84.3 (84.1)	94.0 (93.7)	94.8 (94.1)	56.7 (56.1)	75.3 (74.8)
$\text{LERT}_{\text{base}}$	80.2 (79.5)	**79.8 (79.3)**	89.5 (89.2)	86.6 (86.4)	85.9 (**85.6**)	85.1 (**84.9**)	94.9 (94.7)	**95.9 (95.2)**	**57.5 (57.1)**	**78.2 (77.5)**
$\text{RoBERTa}_{\text{large}}$	82.1 (81.3)	81.2 (80.6)	90.4 (90.0)	87.0 (86.8)	86.3 (85.7)	**85.8** (84.9)	**95.8** (94.9)	95.8 (94.9)	58.8 (58.4)	78.5 (78.2)
$\text{ELECTRA}_{\text{large}}$	81.5 (80.8)	81.0 (**80.9**)	**90.7 (90.4)**	87.3 (**87.2**)	**86.7 (86.2)**	85.1 (84.8)	95.2 (94.6)	95.3 (94.8)	57.2 (56.9)	78.8 (78.4)
$\text{MacBERT}_{\text{large}}$	**82.4 (81.8)**	**81.3** (80.6)	90.6 (90.3)	**87.6** (87.1)	86.2 (85.7)	85.6 (**85.0**)	95.7 (**95.0**)	95.9 (95.1)	**59.0 (58.8)**	79.0 (78.7)
$\text{PERT}_{\text{large}}$	81.0 (80.4)	80.4 (80.1)	90.0 (89.7)	87.2 (86.9)	86.3 (85.8)	85.0 (84.8)	94.5 (94.0)	95.3 (94.8)	57.4 (57.2)	78.1 (77.8)
$\text{LERT}_{\text{large}}$	81.7 (81.2)	81.0 (80.7)	90.2 (90.0)	87.3 (86.9)	86.6 (86.0)	85.1 (84.7)	95.6 (94.9)	**96.2 (95.4)**	58.7 (58.5)	**79.4 (78.9)**

Table 5: Experimental results on text classification tasks (including natural language inference tasks): XNLI, LCQMC, BQ Corpus, ChnSentiCorp, TNEWS, and OCNLI.

System	MSRA-NER	People's Daily
$\text{BERT}_{\text{base}}$	95.3 (94.9)	95.3 (95.1)
$\text{RoBERTa}_{\text{base}}$	95.5 (95.1)	95.1 (94.9)
$\text{ELECTRA}_{\text{base}}$	95.4 (95.0)	95.1 (94.9)
$\text{MacBERT}_{\text{base}}$	95.3 (95.1)	95.2 (94.9)
$\text{PERT}_{\text{base}}$	95.6 (95.3)	95.3 (95.1)
$\text{LERT}_{\text{base}}$	**95.7** (95.4)	**95.6** (95.4)
$\text{RoBERTa}_{\text{large}}$	95.5 (95.5)	95.7 (95.4)
$\text{ELECTRA}_{\text{large}}$	95.0 (94.8)	94.9 (94.8)
$\text{MacBERT}_{\text{large}}$	96.2 (95.9)	95.8 (95.7)
$\text{PERT}_{\text{large}}$	96.2 (**96.0**)	96.1 (95.8)
$\text{LERT}_{\text{large}}$	**96.3** (**96.0**)	**96.3** (**96.0**)

Table 6: Experimental results (F-score) on NER tasks.

We use a universal initial learning rate for each task for the same model size, with 5e-5 for small-sized models, 3e-5 for base-sized models, and 2e-5 for large-sized models. Other details for task fine-tuning are shown in Table 3. The implementations are based on original BERT.[5]

4.3 Main Results

We mainly compare our results with pre-trained language models that use a similar amount of training data. Experimental results on base-sized and large-sized models are shown in Table 4, 5, and 6.

For machine reading comprehension tasks, LERT yields significant improvements over various pre-trained language models by a large margin on both base-sized and large-sized LERT. This indicates that LERT can better handle complex task that requires various types of linguistic knowledge (e.g., machine reading comprehension).

For text classification tasks, the results are varied. We can see that LERT yields the best performance on several tasks, such as TNEWS, OCNLI, ChnSentiCorp, etc. For other tasks, the average results are competitive against the best-performing pre-trained language model. Unlike the MRC task, the text classification tasks are usually determined by very few words in the input sequence, such as sentiment words, negation words, etc. In this context, we speculate that the additional linguistic knowledge introduced in LERT is not that useful for further improving classification accuracy. Nonetheless, we can see that LERT still provides decent scores on several classification tasks.

For named entity recognition tasks, we can see that LERT yields the best performance on both tasks, including base-sized and large-sized variants. The results are expected because the NER task

[5] https://github.com/google-research/bert

System	Param	CMRC 2018		DRCD		XNLI	LC	BQ	CSC	TN	OC	MSRA	PD
		EM	F1	EM	F1	ACC	ACC	ACC	ACC	ACC	ACC	F	F
RBT3	38M	62.2	81.8	75.0	83.9	72.3	85.1	83.3	92.8	-	-	-	-
$\text{ELT}_{\text{small+}}$	12M	**68.5**	**85.2**	82.9	88.7	74.6	85.8	82.1	93.6	-	-	-	-
$\text{ELT}_{\text{small}}$	12M	*67.8*	83.4	79.0	85.8	73.1	***85.9***	82.0	***94.3***	-	-	-	-
$\text{BERT}_{\text{small}}$	15M	65.3	83.9	81.7	88.1	74.6	85.7	83.0	93.6	***55.2***	70.8	91.8	91.5
$\text{LERT}_{\text{small}}$	15M	*67.8*	***85.2***	***83.2***	***89.4***	***75.2***	85.3	***83.4***	94.0	54.9	***71.0***	***92.3***	***92.1***

Table 7: Test results on small models. ELT: ELECTRA. Overall best scores are depicted in boldface, and the comparable best score (the same training data size) is shown in italics.

System	CMRC		DRCD		XNLI	LC	BQ	CSC	TN	OC	MSRA	PD	Average
	EM	F1	EM	F1	ACC	ACC	ACC	ACC	ACC	ACC	F	F	
Baseline	66.8	86.7	89.0	94.1	78.1	88.7	85.1	94.2	56.4	76.0	94.6	94.4	83.58
+ POS	67.0	86.9	89.0	93.9	78.4	88.9	85.1	94.2	56.3	75.6	95.1	95.2	83.72 (+0.14)
+ NER	67.0	87.2	89.4	94.3	78.5	89.1	85.3	94.2	56.8	76.0	95.5	95.4	83.97 (+0.39)
+ DEP	67.2	86.9	89.3	94.2	78.6	88.9	85.0	94.0	56.0	76.0	94.9	95.0	83.72 (+0.14)
+ All	67.5	87.3	89.5	94.4	78.8	88.9	85.2	94.2	56.7	76.4	95.4	95.4	84.03 (+0.45)
$\text{LERT}_{\text{base}}$	68.4	87.9	90.2	94.9	79.5	89.2	85.6	94.7	57.1	77.5	95.4	95.4	84.65 (+1.07)

Table 8: Ablation results on using different linguistic features. We report five-run average scores on the development set of each task. The reported results are on *base*-sized PLMs, trained with 500k steps (except for $\text{LERT}_{\text{base}}$).

is added into the pre-training stage as one of the linguistic tasks. Further analyses are presented in Section 5.

Overall, LERT yields significant improvements on MRC and NER tasks and achieves competitive performance on TC tasks over various pre-trained language models.

4.4 Results on Small Models

Along with conventional base-sized and large-sized LERT, we also train a small-sized $\text{LERT}_{\text{small}}$. Unlike base-sized and large-sized models, small models are usually not comparable to the previous work due to various types of model structure, including hidden size, number of layers, number of attention heads, etc. In this paper, to make the results comparable, we also train a $\text{BERT}_{\text{small}}$, which shares the same training recipe with $\text{LERT}_{\text{small}}$, except that it is only trained on the MLM task. The experimental results are shown in Table 7.

Similar to the base-sized and large-sized models, LERT yields consistent improvements on MRC and NER tasks and moderate improvements on classification tasks, which further demonstrates that linguistic knowledge preference for different tasks differs. We also compare $\text{LERT}_{\text{small}}$ with RBT3 (38M parameters), $\text{ELECTRA}_{\text{small}}$ (12M), and $\text{ELECTRA}_{\text{small+}}$ (12M). Note that ELECTRA (Clark et al., 2020) uses embedding decomposition that projects word embedding into a smaller vector and then uses a fully-connected layer to project into hidden size. However, LERT does not apply this approach, resulting in a little bit more parameter sizes than ELECTRA. The results show that $\text{LERT}_{\text{small}}$ performs better on a majority of downstream tasks over $\text{ELECTRA}_{\text{small}}$ and RBT3, and even better than $\text{ELECTRA}_{\text{small+}}$, which was trained on 180G pre-training data.

5 Analysis

5.1 Ablation Study

In order to identify the effectiveness of each linguistic task, in this section, we perform the ablation study on LERT. We add each linguistic task on top of MLM (baseline) to verify their effectiveness individually.[6] The results are shown in Table 8. As we can see that all three types of linguistic features contribute to the overall improvement positively, where the NER features are the most important, especially for downstream NER tasks. Furthermore, using all three linguistic features yields another boost in the final performance, where all downstream tasks yield consistent improvements.

5.2 Effect of Linguistic Task Order

For better pre-training LERT, we propose a linguistically-informed pre-training strategy, which

[6]We do not perform task warmup for these experiments to keep comparisons as pure as possible.

learns basic linguistic knowledge faster (POS) than the high-level knowledge (NER and DEP), forming a "PND" scheme.[7] To further demonstrate its effectiveness, we also trained another three $\text{LERT}_{\text{base}}$ models that use different task warmup strategies, including PDN, NPD, and DNP, which indicates the different warmup order of linguistic tasks. To demonstrate the effectiveness of the task warmup strategy, we add another "no warmup" experiment that uses Equation 6 for training. The results are shown in Figure 3. Overall, the original implementation in LERT (i.e., "PND" scheme) yields the best performance among all variants, where we conclude our findings as follows.

- By comparing "PND", "PDN", and "NPD", we discover that results are better when "POS" feature learns faster, which matches our intuitive thinking that fundamental knowledge should be learned faster.
- "NPD" scheme yields the best performance on NER tasks, suggesting that if the pre-trained task is directly associated with the downstream tasks, it is always better to learn from the beginning.
- "PND" and "PDN" yield better overall performance than the others, indicating that faster basic knowledge (POS) learning is helpful for better high-level knowledge learning.
- All task warmup schemes show better overall performance than no task warmup strategy (i.e., equal weights for each task).

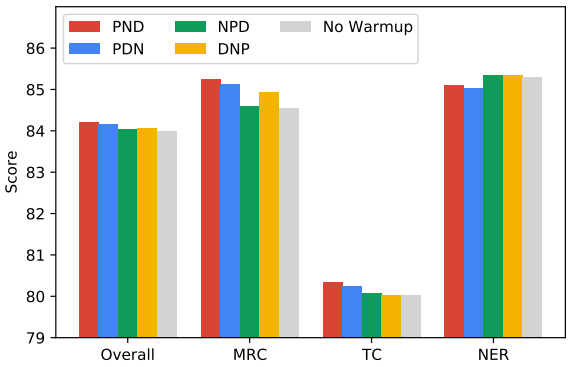

Figure 3: Effect of different linguistic task order in LIP. Note that the scores of NER are subtracted by 10 for clarity. All models are trained with 1M steps.

[7]We use the initials of each linguistic task to denote a specific warmup scheme. Concretely, "PND" means the POS features learn fastest, then NER, followed by DEP.

5.3 Effect of Linguistical Masking

In this paper, we use linguistical multi-task learning to formalize LERT, where the linguistic knowledge is used at the output as labels in the pre-training stage. However, we wonder if linguistic knowledge can be applied at the input as a hint of masking and whether it is more effective than LERT's implementation.

To achieve this goal, we extend the original masked language model as a linguistic masked language model (LMLM). In traditional MLM, the masking token is `[MASK]`, which does not carry any linguistic information. In LMLM, we further incorporate linguistic tags into `[MASK]`, forming a list of different masking tokens. For example, if the POS tag of the masked token is a noun, then the corresponding masked token is set as `[MASK-POS-n]`. In this way, the model is informed with additional linguistic hints for the masked tokens. We train three types of LMLM w.r.t. each type of linguistic knowledge. Also, we set up two additional settings, called "All" (incorporating all three types of linguistic tags into `[MASK]`) and "Mix" (randomly assign one linguistic tag into `[MASK]`. We use the same training settings as in Table 8. The results of LMLM are listed in Table 9.

As we can see that incorporating NER tags into the masked token yields improvement over vanilla MLM, especially for the NER tasks. However, for most of the other settings, LMLM does not yield consistent improvements. By comparing to the results in Table 8, we can see that exploiting linguistic knowledge as the training target yields consistent and significant improvements over vanilla MLM and LMLMs. For example, using NER tags as the training target (average score: 83.97, in Table 8) yields better performance than in LMLM (average score: 83.70). These results indicate that the design of LERT is valid.

System	MRC	TC	NER	Average
MLM	84.2	79.8	94.5	83.58
+ POS-mask	83.1	79.4	94.6	83.14
+ NER-mask	83.9	79.9	94.8	83.70
+ DEP-mask	83.7	79.7	94.5	83.43
+ All-mask	83.5	79.5	94.5	83.26
+ Mix-mask	83.9	79.5	94.4	83.40

Table 9: Results of linguistical masking (LMLM).

6 Conclusion

In this paper, we propose a new pre-trained language model called LERT, which directly incorporates three types of linguistic features and performs multi-task pre-training along with the masked language model. Three types of common linguistic features, including POS, NER, and DEP, are generated by LTP, and LERT learns to predict both the original word and its linguistic tags for masked tokens. To better acquire linguistic knowledge, we also propose a linguistically-informed pre-training strategy that learns basic linguistic faster than the high-level ones, which we empirically find useful. We carried out extensive and robust experiments on ten Chinese natural language understanding tasks. The experimental results show that LERT could bring significant improvements over various comparable pre-trained language models, demonstrating that linguistic knowledge can still boost the performance of pre-trained language models, especially for small-sized models.

In the future, we are going to incorporate more types of linguistic features into pre-trained language models, such as semantic dependency parsing, etc. Also, as the proposed task warmup strategy seems to be generally useful, we are going to investigate if it is helpful to other multi-task learning scenarios.

Acknowledgments

Yiming Cui would like to thank continuous support from Google's TPU Research Cloud (TRC) program for Cloud TPU access.

References

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 13–16. Association for Computational Linguistics.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4946–4951, Brussels, Belgium. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. A Span-Extraction Dataset for Chinese Machine Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5886–5891, Hong Kong, China. Association for Computational Linguistics.

Yiming Cui, Ziqing Yang, and Ting Liu. 2022. Pert: Pre-training bert with permuted language model. *arXiv preprint arXiv:2203.06906*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kuebler, and Larry Moss. 2020. Ocnli: Original chinese natural language inference. In *Findings of EMNLP*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, pages 1–17.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced Chinese sequence labeling using BERT adapter. 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 5847–5858, Online. Association for Computational Linguistics.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

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*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018. Drcd: a chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*.

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

Songbo Tan and Jin Zhang. 2008. An empirical study of sentiment analysis for chinese documents. *Expert Systems with applications*, 34(4):2622–2629.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

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.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. 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 5412–5422, Online. Association for Computational Linguistics.

Taolin Zhang, Junwei DOng, Jianing Wang, Chengyu Wang, Ang Wang, Yinghui Liu, Jun Huang, Yong Li, and Xiaofeng He. 2022. Revisiting and advancing chinese natural language understanding with accelerated heterogeneous knowledge pre-training. *arXiv preprint arXiv:2210.05287*.

Zhuosheng Zhang, Hanqing Zhang, Keming Chen, Yuhang Guo, Jingyun Hua, Yulong Wang, and Ming Zhou. 2021. Mengzi: Towards lightweight yet ingenious pre-trained models for chinese. *arXiv preprint arXiv:2110.06696*.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2020. LIMIT-BERT : Linguistics informed multi-task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online. Association for Computational Linguistics.