Building Chinese Biomedical Language Models via Multi-Level Text Discrimination

Anonymous ACL submission

Abstract

Pre-trained language models (PLMs), such as BERT and GPT, have revolutionized the field of NLP, not only in the general domain but also in the biomedical domain. Most prior efforts in building biomedical PLMs have resorted simply to domain adaptation and focused mainly on English. In this work we introduce eHealth, a Chinese biomedical PLM built from scratch with a new pre-training framework. This new framework pre-trains eHealth as a discriminator through both token- and sequence-level discrimination. The former is to detect input tokens corrupted by a generator and recover their original identities from plausible candidates, while the latter is to further distinguish corruptions of a same original sequence from those of others. As such, eHealth can learn language semantics at both token and sequence levels. Extensive experiments on 11 Chinese biomedical language understanding tasks of various forms verify the effectiveness and superiority of our approach. We release the pre-trained model to the public,[1] and will also release the code later.

1 Introduction

Pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) and its variants (Yang et al., 2019; Liu et al., 2019) have revolutionized the field of NLP, establishing new state-of-the-art on conventional language understanding and generation tasks. Following the great success in the general domain, researchers have started to investigate building domain-specific PLMs in highly specialized domains, *e.g.*, science (Beltagy et al., 2019), law (Chalkidis et al., 2020), or finance (Liu et al., 2020). Biomedicine and healthcare, as a field with large, rapidly growing volume of free text and continually increasing demand for text mining, has received massive attention and achieved rapid progress.

Biomedical PLMs are typically built by adapting a general-domain PLM to the biomedical domain with (almost) the same model architecture and training objectives, as exemplified by BioBERT (Lee et al., 2020), PubMedBERT (Gu et al., 2020), and BioELECTRA (Kanakarajan et al., 2021). This domain adaptation is achieved via either *continual pre-training* on in-domain text (Gururangan et al., 2020), or *pre-training from scratch* further with an in-domain vocabulary (Gu et al., 2020; Lewis et al., 2020b), which has shown to be particularly useful for English biomedical text understanding.

As for the Chinese biomedical field, MC-BERT (Zhang et al., 2020) and PCL-MedBERT are two initial attempts that continually pre-train a general-domain BERT on in-domain text. But unfortunately they fail to achieve satisfactory performance compared with their general-domain rivals (Zhang et al., 2021a). SMedBERT (Zhang et al., 2021b) and EMBERT (Cai et al., 2021) also continually pre-train from the general-domain BERT, but in knowledge-enhanced fashions. These two models rely on external (and often private) knowledge and have not been released to the public yet. So far there is still a lack of publicly available, high-quality biomedical PLMs in Chinese.

In this paper we present **eHealth**, a Chinese language representation model pre-trained over large-scale biomedical text corpora. Unlike most previous studies that simply resort to direct domain adaptation, we build eHealth with a new self-supervised learning framework, which, similar to ELECTRA (Clark et al., 2020), consists of a discriminator and a generator. The generator is to produce corrupted input, and the discriminator, as the final target encoder, is trained via multi-level text discrimination. Specifically, we employ (i) *token-level discrimination* that discriminates corrupted tokens from original ones, and (ii) *sequence-level discrimination* that further discriminates corruptions of a same original sequence from those of others in a contrastive learning fashion (Chen et al., 2020). This multi-level discrimination enables eHealth to learn language

[1] AnonymousURL

semantics at both token and sequence levels.

As a new Chinese biomedical PLM, eHealth has two distinguishing features: built-from-scratch and easy-to-deploy. By the former we mean that unlike all prior arts that start pre-training from a general-domain Chinese BERT and directly use the associated vocabulary, eHealth is pre-trained entirely from scratch with a newly built in-domain vocabulary. This vocabulary, as we will show later in our experiments, can better tokenize biomedical text and may lead to better understanding of such text. And by the latter we mean that eHealth relies solely on the text itself, requiring no additional retrieval, linking, or encoding of relevant knowledge as those knowledge-enhanced models do, and thereby could be applied rather easily during fine-tuning.

We evaluate eHealth on 11 diversified Chinese biomedical language understanding tasks, including (i) the 8 tasks of text classification and matching, medical information extraction, and medical term normalization from the CBLUE benchmark (Zhang et al., 2021a), and (ii) another 3 medical question answering tasks cMedQNLI (Zhang et al., 2020), webMedQA (He et al., 2019), and NLPEC (Li et al., 2020). Experimental results reveal that eHealth, as a standard base-sized model pre-trained from scratch on biomedical corpora, consistently outperforms previous state-of-the-art PLMs in almost all cases, no matter those from the general domain or biomedical domain, and no matter those base-sized or even large-sized.

The main contributions of this work are two-fold. Firstly, we propose a new Chinese biomedical PLM and release the pre-trained model to the public. This new model shows superior ability in Chinese biomedical text understanding and is easy to deploy. Secondly, we devise a new algorithm for language model pre-training and verify its effectiveness in the biomedical domain. This pre-training algorithm is quite generic and may be readily adapted to other domains beyond biomedicine. We leave such exploration open to future work.

2 Background

Before diving into the details of our approach, we briefly discuss related studies on building PLMs in general and biomedical domains.

General Domain PLMs. Recent years have seen remarkable success of PLMs in the field of NLP. These PLMs are typically built with self-supervised learning over massive unlabeled text in the general domain, *e.g.*, Wikipedia, newswire, or Web articles (Radford et al., 2018). *Masked language modeling* (MLM), which trains a model to recover the identities of a small subset of masked-out tokens (typically 15%), is the most prevailing self-supervised objective, first introduced in BERT (Devlin et al., 2019) and then widely adopted by follow-up studies (Liu et al., 2019; Lan et al., 2020; Joshi et al., 2020; Sun et al., 2020). Despite their effectiveness and popularity, MLM-based approaches can only learn from those 15% masked-out tokens per input, and therefore incur high compute costs.

To address this low efficiency issue, ELECTRA (Clark et al., 2020) uses a new pre-training framework. Specifically, it corrupts an input sequence by replacing some of the tokens with plausible alternatives sampled from an auxiliary generator, and trains a discriminator to predict for each token in that sequence whether it is original or replaced, *i.e.*, *replaced token detection* (RTD). As the discriminator can learn from all input tokens rather than just 15% of them, ELECTRA enjoys better efficiency and accelerates training.

While achieving empirical success, there are concerns about whether the over-simplified RTD task of ELECTRA, as a binary classification problem, is informative enough for language modeling (Aroca-Ouellette and Rudzicz, 2020). Xu et al. (2020) and Shen et al. (2021) thus proposed training the model via a generalization of RTD while a simplification of MLM, by recovering for each token its original identity from a few plausible candidates, rather than from the whole vocabulary.

Another limitation of ELECTRA is that it is pre-trained solely at the token level but lacks semantics at the sequence level. Incorporating sequence level signals, *e.g.*, next sentence prediction (Devlin et al., 2019), sentence order prediction (Lan et al., 2020), and sentence contrastive learning (Fang et al., 2020; Meng et al., 2021), has been widely accepted in the community and shown to be beneficial in specific tasks (Lewis et al., 2020a; Guu et al., 2020).

In this paper, to build a Chinese biomedical PLM, we employ the ELECTRA framework which favors the efficiency of pre-training. Within this framework, we strengthen the oversimplified RTD task and introduce sequence-level signals, which further improves the quality of pre-training.

Biomedical PLMs. Continual pre-training is perhaps the most straightforward way to build biomedical PLMs, in which the model weights are initial-

ized from a well-trained general-domain model and the same vocabulary is used (Alsentzer et al., 2019; Lee et al., 2020). Also, there are findings showing that pre-training from scratch using domain specific data along with domain specific vocabulary would bring further improvements, particularly in English (Gu et al., 2020; Lewis et al., 2020b). Early attempts focused on adapting BERT, while recent studies have switched to its modern variants like RoBERTa, ALBERT, and ELECTRA (Kanakarajan et al., 2021; Alrowili and Shanker, 2021).

While great efforts have been made to build English biomedical PLMs, there is only a few studies discussing building biomedical PLMs in Chinese, *e.g.*, MC-BERT (Zhang et al., 2020), SMedBERT (Zhang et al., 2021b), and EMBERT (Cai et al., 2021), all resumed from a general-domain BERT, with the latter two further in knowledge-enhanced fashions.[2] Models like this typically require extra knowledge and consequently the retrieval, linking, and encoding of such knowledge. They are not that easy to be applied to downstream tasks.

3 Methodology

This section presents eHealth, a Chinese language model pre-trained from biomedical text. It in general follows the generator-discriminator framework of ELECTRA, where the generator G is introduced to construct pre-training signals and the discriminator D is used as the final target encoder. But unlike ELECTRA that merely adopts a token-level binary classification to train the discriminator, we train it with (i) a more informative token-level discrimination, and (ii) another sequence-level discrimination. The overview of eHealth is illustrated in Figure 1.

3.1 Generator

The generator G is a Transformer encoder (Vaswani et al., 2017) trained by *masked language modeling* (MLM). Given an input sequence $\mathbf{x} = [x_1, \cdots, x_n]$, it first selects a random set of positions to mask out and replaces tokens at these positions with a special symbol `[MASK]`.[3] This masked sequence, denoted as $\mathbf{x}^M$, is then passed into the Transformer encoder to produce contextualized representations $h_G(\mathbf{x}^M)$, and thereafter a softmax layer to predict

[2] Actually there are two versions of EMBERT, one initialized with BERT and the other with MC-BERT, which is also resumed from BERT.

[3] Typically 15% of the tokens are masked out, among which 80% are replaced with `[MASK]`, 10% replaced with a random token, and 10% kept unchanged.

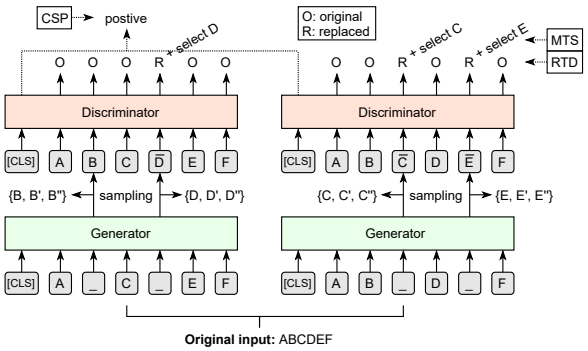

Figure 1: Overview of eHealth. Each input sequence is corrupted twice independently by the generator. These two corruptions are fed into the discriminator for replaced token detection (RTD) and multi-token selection (MTS), *i.e.*, token-level discrimination. And they also form a positive pair for contrastive sequence prediction (CSP), *i.e.*, sequence-level discrimination.

the original identities of those masked-out tokens:

$$p_G(x_t|\mathbf{x}^M) = \frac{\exp\left(e(x_t)^{\mathrm{T}} h_G(\mathbf{x}^M)_t\right)}{\sum_{x' \in V} \exp\left(e(x')^{\mathrm{T}} h_G(\mathbf{x}^M)_t\right)}. \quad (1)$$

Here, $p_G(x_t|\mathbf{x}^M)$ is the probability that G predicts token x_t appears at the t-th masked position in $\mathbf{x}^M$, $h_G(\mathbf{x}^M)_t$ the contextualized representation for that position, $e(\cdot)$ the embedding lookup operation on each token, and V the vocabulary of all tokens. The corresponding loss function is:

$$\mathcal{L}_{\mathrm{MLM}}(\mathbf{x}, \mathbf{x}^M; G) = \sum_{t: x_t^M = \texttt{[MASK]}} -\log p_G(x_t|\mathbf{x}^M), \quad (2)$$

where the summation is taken only over the masked positions. The generator is used to construct pre-training signals for the discriminator, and will be discarded after pre-training.

3.2 Discriminator

The discriminator D, as our final target encoder, is also a Transformer architecture. It takes as input corrupted sequences constructed by the generator, and is trained through two-level text discrimination, *i.e.*, token-level and sequence-level, so as to encode language semantics at both levels.

Token-Level Discrimination. We consider two token-level tasks: *replaced token detection* (RTD) and *multi-token selection* (MTS). RTD is the standard pre-training task of ELECTRA, which detects replaced tokens in a corrupted sequence, and MTS further selects original identities for those replaced tokens. Specifically, given input sequence $\mathbf{x}$ and its masked version $\mathbf{x}^M$, for each masked position t,

we sample a token from the generator's prediction $\hat{x}_t \sim p_G(x_t|\mathbf{x}^M)$ (cf. Eq. (1)), replace the original token x_t with $\hat{x}_t$, and create a corrupted sequence $\mathbf{x}^R$. We also create a set of candidate tokens, denoted as S_t, for each masked position t, by drawing k non-original tokens from $p_G(x_t|\mathbf{x}^M)$ along with the original token x_t. The discriminator D encodes the corrupted sequence $\mathbf{x}^R$ and produces contextualized representations $h_D(\mathbf{x}^R)$.

RTD learns to discriminate whether each token in $\mathbf{x}^R$ is original or replaced, *i.e.*, coming from the true data distribution or the generator distribution. It uses a sigmoid layer on top of $h_D(\mathbf{x}^R)$ to perform this binary classification, where the probability that x_t^R matches the original token x_t is determined as:

$$p_D(x_t^R = x_t) = \frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}} h_D(\mathbf{x}^R)_t)}, \quad (3)$$

and the corresponding loss function is:

$$\mathcal{L}_{\mathrm{RTD}}(\mathbf{x}, \mathbf{x}^R; D) = \sum_{t=1}^{n} \big[-\mathbb{1}(x_t^R = x_t) \log p_D(x_t^R = x_t) - \mathbb{1}(x_t^R \neq x_t) \log(1 - p_D(x_t^R = x_t)) \big]. \quad (4)$$

As merely a binary classification task, RTD might not be informative enough for language modeling.

MTS strengthens RTD by training the discriminator to further recover original identities of those replaced tokens. For each position t where the token is replaced, *i.e.*, $x_t^R \neq x_t$, MTS corrects the token and recovers its original identity from candidate set S_t. The probability of picking the original identity x_t out of S_t for the correction is:

$$p_D(x_t|\mathbf{x}^R, S_t) = \frac{\exp\left(e(x_t)^{\mathrm{T}} h_D(\mathbf{x}^R)_t\right)}{\sum_{x' \in S_t} \exp\left(e(x')^{\mathrm{T}} h_D(\mathbf{x}^R)_t\right)}, \quad (5)$$

where $e(\cdot)$ is again the embedding lookup operation. The loss function is defined as:

$$\mathcal{L}_{\mathrm{MTS}}(\mathbf{x}, \mathbf{x}^R, \mathcal{S}; D) = \sum_{t: x_t^R \neq x_t} -\log p_D(x_t|\mathbf{x}^R, S_t), \quad (6)$$

where $\mathcal{S} = \{S_t\}_{t: x_t^R \neq x_t}$ is a collection of candidate sets at all positions with replaced tokens, and the summation is taken only over these positions. MTS is essentially a $(k+1)$-class classification problem. It is more challenging than RTD and hence pushes the discriminator to learn representations that encode richer semantic information (Xu et al., 2020; Shen et al., 2021).

Sequence-Level Discrimination. Besides token-level tasks, we consider a sequence-level task in addition, *i.e.*, *contrastive sequence prediction* (CSP) which learns to discriminate corruptions of a single original sequence from those of the others. CSP employs a classic contrastive learning framework (Chen et al., 2020). Specifically, for each original input sequence we create two corrupted versions, each by independently picking some random positions to mask out and filling the masked positions with samples from the generator, just like how we do in token-level discrimination as described above. The two corruptions of a same original sequence $\mathbf{x}$, denoted as $\mathbf{x}_i^R$ and $\mathbf{x}_j^R$, are taken as a positive pair, and corruptions of other sequences within the same minibatch as $\mathbf{x}$ are regarded as negative examples, the set of which is denoted as $N(\mathbf{x})$. The CSP task is then to identify $\mathbf{x}_j^R$ in $N(\mathbf{x})$ for a given $\mathbf{x}_i^R$, and the contrastive loss is accordingly defined as:

$$\mathcal{L}_{\mathrm{CSP}}(\mathbf{x}, \mathbf{x}_i^R, \mathbf{x}_j^R; D) = -\log \frac{\exp\big(\mathrm{s}(\mathbf{x}_i^R, \mathbf{x}_j^R)/\tau\big)}{\sum_{\mathbf{x}_k^R \in N(\mathbf{x})} \exp\big(\mathrm{s}(\mathbf{x}_i^R, \mathbf{x}_k^R)/\tau\big)}, \quad (7)$$

where $\mathrm{s}(\cdot,\cdot)$ is the similarity measure between two sequences and τ is a temperature hyperparameter. We represent each sequence by the ℓ_2-normalized representation of its `[CLS]` token, *i.e.*, $\mu_D(\cdot) = h_D(\cdot)_1 / \|h_D(\cdot)_1\|$ where $h_D(\cdot)_1$ stands for the representation of the first token in a sequence output by the discriminator D, and determine the similarity as $\mathrm{s}(\mathbf{u}, \mathbf{v}) = \mu_D(\mathbf{u})^{\mathrm{T}} \mu_D(\mathbf{v})$. This contrastive learning task requires $\mathbf{x}_i^R$ and $\mathbf{x}_j^R$ to stay close to each other while away from other corrupted sequences in the same minibatch, and therefore encourages the discriminator to learn representations invariant to token-level alterations. A similar task has been considered recently by Meng et al. (2021) to help build general-domain PLMs, but it uses a different data transformation procedure to generate positive pairs by random cropping, resulting in asymmetric encoding of sequence pairs.

3.3 Model Training

Putting the generator and discriminator as well as their associated tasks together, we train eHealth by minimizing the following combined loss:

$$\min_{G,D} \ \mathcal{L}_{\mathrm{MLM}} + \lambda_1 \mathcal{L}_{\mathrm{RTD}} + \lambda_2 \mathcal{L}_{\mathrm{MTS}} + \lambda_3 \mathcal{L}_{\mathrm{CSP}}. \quad (8)$$

The first term is a generator loss, and the latter three are discriminator losses which are not propagated through the generator. $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters balancing these loss terms. After pre-training, we throw out the generator and fine-tune only the discriminator on downstream tasks.

4 Experiments

This section first describes our experimental setups for pre-training and fine-tuning, and then presents evaluation results and further ablation.

4.1 Pre-training Setups

Pre-training Data. We use four Chinese datasets for pre-training: (i) *Dialogues* consisting of about 100 million de-identified doctor-patient dialogues from online healthcare services; (ii) *Articles* consisting of about 6.5 million popular scientific articles on medicine and healthcare oriented to the general public; (iii) *EMRs* consisting of about 6.5 million de-identified electronic medical records from specific hospitals; and (iv) *Textbooks* consisting of about 1,500 electronic textbooks on medicine and clinical pathology. The contents of these datasets are quite diversified, covering most aspects of biomedicine, namely scientific, clinical, and consumer health (Jin et al., 2021). After collecting raw text, we conduct minimum pre-processing of deduplication and denoising on each of the four datasets. We then tokenize the text using a newly built in-domain vocabulary (detailed later). Sequences longer than 512 tokens are segmented into shorter chunks according to sentence boundaries, and those shorter than 32 tokens are discarded. Table 1 summarizes the datasets used for pre-training.

In-domain Vocabulary. Unlike previous studies that continually pre-train from and thereby use the vocabulary of a general-domain Chinese BERT, we train eHealth *from scratch* with its own in-domain vocabulary built specifically for Chinese biomedical text. Gu et al. (2020) have shown that training from scratch with an in-domain vocabulary is a better choice than continue pre-training while building English biomedical PLMs, primarily because the in-domain vocabulary can better handle highly specialized biomedical terms. This, however, has never been investigated in the Chinese biomedical field. To build the in-domain vocabulary, we randomly sample 1M documents from the pre-training data, convert all characters to lowercase, normalize special Unicodes like half-width characters or enclosed alphanumerics, and split Chinese characters, digits, and emoji Unicodes. Then we use the open-source implementation from the Tensor2Tensor library[4] to create a WordPiece vocabulary (Wu et al., 2016). We throw out tokens appearing less than 5 times and keep the vocabulary of size to about 20K tokens, which is similar to the general-domain Chinese BERT. Table 2 compares tokenization results obtained by (i) the original vocabulary of standard BERT and (ii) our newly built in-domain vocabulary. We can see that as both the two vocabularies are mainly based on single Chinese characters, the differences between them are not that significant as in English. But still the in-domain vocabulary works pretty better on abbreviations of specialized biomedical terms, including not only those rare ones like IHC (immunohistochemistry) and TSHR (thyroid stimulating hormone receptor), but also those relatively popular ones like AIDS (acquired immune deficiency syndrome) and MRI (magnetic resonance imaging).

[4] https://github.com/tensorflow/tensor2tensor

Corpus	Size	# Tokens	Sub-domain
Dialogues	94.6GB	31.1B	consumer health
Articles	11.2GB	3.5B	consumer health
EMRs	16.0GB	4.5B	clinical
Textbooks	5.1GB	1.6B	scientific
Total	126.9GB	40.7B	N/A

Table 1: Corpora used for eHealth pre-training.

免疫组化IHC测定TSHR阳性 (Positive expression of TSHR by immunohistochemistry (IHC))
BERT: 免, 疫, 组, 化, **i**, **##hc**, 测, 定, **ts**, **##hr**, 阳, 性
eHealth: 免, 疫, 组, 化, **ihc**, 测, 定, **tshr**, 阳, 性
ECOG评分4分者 (Those with ECOG score of 4)
BERT: **eco**, **##g**, 评, 分, 4, 分, 者
eHealth: **ecog**, 评, 分, 4, 分, 者
但不包括HIV/AIDS (But excluding HIV/AIDS)
BERT: 但, 不, 包, 括, hiv, /, **ai**, **##ds**
eHealth: 但, 不, 包, 括, hiv, /, **aids**
胸部增强CT及头颅MRI (Enhanced chest CT & skull MRI)
BERT: 胸, 部, 增, 强, ct, 及, 头, 颅, **mr**, **##i**
eHealth: 胸, 部, 增, 强, ct, 及, 头, 颅, **mri**

Table 2: Comparison of tokenization results obtained by BERT and eHealth. Differences highlighted in bold.

Pre-training Configurations. We train eHealth with the standard *base-size* configuration, just like most previous biomedical PLMs. The discriminator gets 12 Transformer layers, each with 12 attention heads, 768 hidden size, and 3072 intermediate size. And we follow Clark et al. (2020) to set the generator 1/3 the size of the discriminator and tie their token and positional embeddings. To generate masked positions, we perform Chinese word segmentation and use the whole word masking strategy

Dataset	Task	Train	Dev	Test	Metric
CMeEE	Named Entity Recognition	15,000	5,000	3,000	Micro-F1
CMeIE	Relation Extraction	14,339	3,585	4,482	Micro-F1
CHIP-CDN	Clinical Term Normalization	6,000	2,000	10,192	Micro-F1
CHIP-CTC	Sentence Classification	22,962	7,682	10,000	Macro-F1
KUAKE-QIC	Sentence Classification	6,931	1,955	1,994	Accuracy
CHIP-STS	Sentence Pair Matching	16,000	4,000	10,000	Macro-F1
KUAKE-QTR	Sentence Pair Matching	24,174	2,913	5,465	Accuracy
KUAKE-QQR	Sentence Pair Matching	15,000	1,600	1,596	Accuracy
cMedQNLI (Zhang et al., 2020)	Question Answer Matching	80,950	9,065	9,969	Micro-F1
webMedQA (He et al., 2019)	Question Answer Matching	252,850	31,605	31,655	Precision@1
NLPEC (Li et al., 2020)	Multiple Choice	18,117	2,500	550	Accuracy

Table 3: Downstream tasks used for evaluation. Tasks in the first group are from CBLUE (Zhang et al., 2021a).

(Cui et al., 2020). We also use dynamic masking with masked positions decided on-the-fly. During pre-training, we mostly follow the hyperparameters recommended by ELECTRA and do not conduct hyperparameter tuning. For newly introduced hyperparameters, we set the loss balancing terms $\lambda_1 = 50$, $\lambda_2 = 20$, $\lambda_3 = 1$ (cf. Eq. (8)), the number of sampled non-original tokens $k = 5$ (cf. Eq. (5)), and temperature $\tau = 0.07$ (cf. Eq. (7)). We train with a batch size of 384 and max sequence length of 512 for 1.65M steps. The full set of pre-training hyperparameters is listed in Appendix A.

4.2 Evaluation Setups

Downstream Tasks. We evaluate on the Chinese Biomedical Language Understanding Evaluation (*CBLUE*) benchmark (Zhang et al., 2021a), which is composed of 8 diversified biomedical NLP tasks, ranging from medical text classification and matching to medical information extraction and medical term normalization. We further consider three medical question answering tasks, namely *cMedQNLI* (Zhang et al., 2020), *webMedQA* (He et al., 2019), and *NLPEC* (Li et al., 2020). The former two are formalized as question-answer matching problems, and the last one a multiple choice problem. Table 3 summarizes the train, dev, test split and metric used for each task. We refer readers to Appendix C and D for further details.

Baseline Models. We compare eHealth against state-of-the-art general-domain Chinese PLMs of: (i) *BERT-base* (Devlin et al., 2019); (ii) *ELECTRA-base/large* (Clark et al., 2020); (iii) *RoBERTa-wwm -ext-base/large* (Liu et al., 2019) trained via MLM with whole word masking strategy; (iv) *MacBERT-base/large* (Cui et al., 2020) trained via improved MLM as a correction task. BERT-base is officially released by Google,[5] and the other models are released by Cui et al. (2020).[6] Besides, we compare to Chinese biomedical PLMs including: (v) *PCL-MedBERT*;[7] (vi) *MC-BERT* (Zhang et al., 2020);[8] (vii) *EMBERT* (Cai et al., 2021); and (viii) *SMedBERT* (Zhang et al., 2021b), all initialized from Google's BERT-base. The full models of EMBERT and SMedBERT are not released to the public, so we just copy the results reported by their authors on medical question answering tasks.

Fine-tuning Configurations. During fine-tuning, we build a lightweight task-specific head on top of the pre-trained encoders for each task. The specific design of these heads is elaborated in Appendix E. For each PLM on each task, we tune the batch size, learning rate, and training epochs in their respective ranges, and determine the optimal setting according to dev performance averaged over three runs with different seeds. The other hyperparameters are set to their default values as in ELECTRA (Clark et al., 2020). The full set of fine-tuning hyperparameters is listed in Appendix B.

4.3 Main Results

Table 4 reports the performance of different PLMs on CBLUE test sets. Note that CBLUE test labels are not released, and one has to submit prediction files to retrieve final scores. To avoid frequent submissions that probe the unseen test labels, we only submit best single run on dev sets for testing. The results show that: (i) The two previous biomedical PLMs, MC-BERT and PCL-MedBERT, indeed per-

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

[6] https://github.com/ymcui/MacBERT

[7] https://code.ihub.org.cn/projects/1775

[8] https://github.com/alibaba-research/ChineseBLUE

Model	CMeEE	CMeIE	CDN	CTC	STS	QIC	QTR	QQR	Avg.
General-domain base-sized models									
BERT-base	66.5	60.6	69.7	68.6	84.7	85.2	59.2	82.5	72.1
ELECTRA-base	65.1	60.4	69.9	67.7	84.4	85.2	61.8	84.0	72.3
MacBERT-base	66.8	61.5	69.7	69.1	84.4	86.0	61.0	83.5	72.7
RoBERTa-wwm-ext-base	66.7	61.4	69.3	68.3	84.2	86.0	60.9	82.7	72.4
General-domain large-sized models									
ELECTRA-large	66.1	59.3	70.8	68.9	85.1	84.1	62.0	85.7	72.8
MacBERT-large	67.6	62.2	70.9	69.7	86.5	85.7	62.5	83.5	73.6
RoBERTa-wwm-ext-large	67.3	62.2	70.6	70.6	85.4	86.7	61.7	86.1	73.8
Biomedical base-sized models									
MC-BERT-base	66.6	60.7	70.1	69.1	85.4	85.3	61.6	82.3	72.6
PCL-MedBERT-base	66.6	60.8	69.9	**70.4**	84.8	85.3	60.2	83.3	72.7
eHealth-base (ours)	**66.9**	**62.1**	**71.9**	69.3	**86.2**	**87.3**	**63.9**	**85.7**	**74.2**

Table 4: Performance (%) of different PLMs on CBLUE test sets. Results generated by the single best run on dev sets. Best scores from **base-sized** models highlighted in bold, and best scores from large-sized models underlined.

Model	cMedQNLI dev | test	webMedQA dev | test	NLPEC dev | test
General-domain base-sized models			
BERT-base	96.4 | 96.4	79.6 | 79.8	67.1 | 54.6
ELECTRA-base	96.0 | 95.9	79.2 | 79.1	69.8 | 54.1
MacBERT-base	96.3 | 96.2	79.9 | 79.8	68.7 | 53.8
RoBERTa-base	96.2 | 96.2	79.7 | 79.9	68.1 | 54.3
General-domain large-sized models			
ELECTRA-large	96.4 | 96.2	80.0 | 80.1	71.8 | 60.0
MacBERT-large	96.3 | 96.3	80.0 | 80.4	70.8 | 56.7
RoBERTa-large	96.3 | 96.2	79.7 | 79.7	71.1 | 56.5
Biomedical base-sized models			
MC-BERT-base	96.4 | 96.5	80.0 | 79.9	68.2 | 54.2
PCL-MedBERT-base	96.3 | 96.2	79.2 | 79.5	67.4 | 52.0
EMBERT†	– | 96.6	– | 80.6	– | –
SMedBERT‡	96.6 | 96.9	79.3 | **81.7**	– | –
eHealth-base (ours)	**97.3** | **97.2**	**80.5** | 80.7	**73.6** | **62.4**

Table 5: Performance (%) of different PLMs on medical QA tasks. RoBERTa-base/large refers to RoBERTa-wwm-ext-base/large. Results marked by † and ‡ copied from original literatures (Cai et al., 2021; Zhang et al., 2021b). Other results produced by ourselves, averaged over best three runs on the dev set of each task. Best scores from **base-sized** models highlighted in bold and best scores from large-sized models underlined.

form better than general-domain BERT-base from which they started continual pre-training, verifying the effectiveness of domain adaptation in building domain-specific language models. However, these two biomedical PLMs fail to surpass some more advanced general-domain PLMs, *e.g.*, MacBERT, of the same model size. (ii) As the model size increases, general-domain large-sized PLMs perform better than those base-sized, *e.g.*, ELECTRA-large, MacBERT-large, and RoBERTa-wwm-ext-large obtain averaged improvements of 0.5%, 0.9%, and 1.4% respectively over their base-sized models. (iii) eHealth, as a base-sized biomedical PLM, outperforms all baseline PLMs in terms of average score, no matter those from the general or biomedical domain, and no matter those base-sized or large-sized. It achieves an average improvement of 1.5% over PCL-MedBERT-base, *i.e.*, the best performing direct opponent of the same model size, and even that of 0.4% over the best performing large-sized model RoBERTa-wwm-ext-large. These results demonstrate the effectiveness and superiority of eHealth in biomedical text understanding.

Table 5 further reports the performance of these PLMs on medical question answering tasks, where scores are averaged over the best three runs selected on the dev split for each task. From the results we can observe similar phenomena as on the CBLUE benchmark. Still eHealth consistently outperforms almost all those PLMs, showing its superior ability in medical question answering.

4.4 Ablation Studies

We provide ablation studies on CBLUE benchmark to show the effects of different pre-training tasks and initialization strategies in eHealth. All variants below are base-sized, trained with the same setting as described in Section 4.1. The only exception is that we train with a smaller batch size of 128 for only 500K steps.

Effects of Pre-training Tasks. The discriminator of eHealth is trained in a multi-task fashion, *i.e.*, (i) token-level discrimination of RTD and MTS and (ii) sequence-level discrimination of CSP. To investigate the effects of different pre-training tasks, we make comparison among: (i) *the full setting* where the discriminator is trained via RTD, MTS, and

Model	CMeEE	CMeIE	CDN	CTC	STS	QIC	QTR	QQR	Avg.
The full setting	**66.56**	**61.62**	70.29	69.58	85.13	**87.46**	**62.00**	**85.53**	**73.52**
w/o CSP	66.47	61.25	69.81	**69.65**	84.61	86.71	61.54	84.52	73.07
w/o MTS	65.76	60.23	**70.43**	68.06	**85.44**	85.61	61.36	84.34	72.65
w/o CSP & MTS	65.56	60.01	70.08	68.46	84.35	86.51	61.08	84.40	72.56
R weights + B vocab	**66.56**	**61.62**	70.29	**69.58**	85.13	**87.46**	62.00	85.53	**73.52**
E weights + E vocab	65.92	61.54	**70.86**	69.53	**85.75**	86.21	**62.38**	**85.59**	73.47
R weights + E vocab	66.33	61.06	70.19	69.50	84.32	87.31	62.33	85.40	73.30

Table 6: Effects of pre-training tasks (top) and initialization strategies (bottom) on CBLUE test sets, where results are generated by single best run on dev sets. All variants are base-sized, trained with batch size 128 for 500K steps. R/B/E in the bottom group stands for R(andom)/B(iomedical)/E(LECTRA), respectively. Within each group **best** scores are highlighted in bold, and second best scores underlined.

CSP; (ii) *w/o CSP* where the sequence-level CSP is removed; (iii) *w/o MTS* where the token-level MTS is removed; and (iv) *w/o CSP & MTS* where both CSP and MTS are removed and thus degenerates to standard ELECTRA pre-training. Table 6 (top) lists the results on CBLUE benchmark, from which we can see that: (i) The full setting performs the best among the four variants, always reporting the best or second best scores on all the 8 diversified tasks. Compared to standard ELECTRA pre-training (w/o CSP & MTS), it achieves an average improvement of 0.96%. This demonstrates the usefulness of our pre-training tasks, in particular CSP and MTS, to build effective PLMs. (ii) No matter CSP or MTS, when applied alone, is able to improve the standard ELECTRA pre-training solely with RTD. Between the two tasks, MTS is, in general, more powerful than CSP. Removing MTS brings an average drop of 0.87% on CBLUE test sets, while removing CSP only brings that of 0.45% on the same benchmark.

Effects of Initialization Strategies. In this work we train eHealth entirely from scratch, with an in-domain vocabulary built specifically for Chinese biomedical text and the model weights randomly initialized. We refer to this strategy as "*R(andom) weights + B(iomedical) vocab*". We compare it to the widely adopted continue pre-training strategy, where model weights are initialized from a general-domain ELECTRA and the associated vocabulary is also used, referred to as "*E(LECTRA) weights + E(LECTRA) vocab*". Besides, to further verify the effects of that in-domain vocabulary, we consider another setting "*R(andom) weights + E(LECTRA) vocab*", where model weights are still randomly initialized but the ELECTRA vocabulary is used. Table 6 (bottom) lists the results on CBLUE benchmark, from which we can see that: (i) Pre-training from scratch with the newly built in-domain vocabulary (R weights + B vocab) overall performs better than continue pre-training (E weights + E vocab), even under a relatively small number of training steps up to 500K.[9] (ii) The improvements mainly come from the in-domain vocabulary. After replacing the vocabulary with that of the general-domain ELECTRA (R weights + E vocab), the overall performance drops from 73.52% to 73.30%.

5 Conclusion

This work presents eHealth, a Chinese biomedical language model pre-trained from in-domain text of de-identified online doctor-patient dialogues, electronic medical records, and textbooks. Unlike most previous studies that directly adapt general-domain PLMs to the biomedical domain, eHealth is trained from scratch with a new self-supervised generator-discriminator framework. The generator is used to produce corrupted input and is discarded after pre-training. The discriminator, as the final encoder, is trained via multi-level discrimination: (i) token-level discrimination that detects input tokens corrupted by the generator and selects original tokens from plausible candidates; and (ii) sequence-level discrimination that further detects corruptions of a same original sequence from those of the others. As such, eHealth can learn language semantics at both levels. Experimental results on CBLUE and 3 medical QA benchmarks demonstrate the effectiveness and superiority of eHealth, which consistently outperforms state-of-the-art PLMs from both the general and biomedical domains. We release our pre-trained model to the public. As the model relies solely on text, it could be applied rather easily during fine-tuning.

[9]The advantage, in fact, will be expanded further as the training step increases according to our initial experiments.

References

Sultan Alrowili and Vijay Shanker. 2021. BioM-Transformers: Building large biomedical language models with BERT, ALBERT and ELECTRA. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 221–227.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Stéphane Aroca-Ouellette and Frank Rudzicz. 2020. On losses for modern language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4970–4981.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3615–3620.

Zerui Cai, Taolin Zhang, Chengyu Wang, and Xiaofeng He. 2021. EMBERT: A pre-trained language model for Chinese medical text mining. In *Web and Big Data*, pages 242–257.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2898–2904.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37-th International Conference on Machine Learning*, pages 1597–1607.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the Eighth International Conference on Learning Representations*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668.

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*, pages 4171–4186.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. CERT: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *arXiv preprint arXiv:2007.15779*.

Tongfeng Guan, Hongying Zan, Xiabing Zhou, Hongfei Xu, and Kunli Zhang. 2020. CMeIE: Construction and evaluation of Chinese medical information extraction dataset. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 270–282.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Junqing He, Mingming Fu, and Manshu Tu. 2019. Applying deep matching networks to Chinese medical question answering: A study and a dataset. *BMC Medical Informatics and Decision Making*, 19(2):91–100.

Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. 2021. Biomedical question answering: A comprehensive review. *arXiv preprint arXiv:2102.05281*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Kamal raj Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. 2021. BioELECTRA: Pretrained biomedical text encoder using discriminators. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 143–154.

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 *Proceedings of the Eighth International Conference on Learning Representations*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020b. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157.

Dongfang Li, Baotian Hu, Qingcai Chen, Weihua Peng, and Anqi Wang. 2020. Towards medical machine reading comprehension with structural knowledge and plain text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1427–1438.

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

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020. FinBERT: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence: Special Track on AI in FinTech*, pages 4513–4519.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. COCO-LM: Correcting and contrasting text sequences for language model pretraining. *arXiv preprint arXiv:2102.08473*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.

Jiaming Shen, Jialu Liu, Tianqi Liu, Cong Yu, and Jiawei Han. 2021. Training ELECTRA augmented with multi-word selection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2475–2486.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8968–8975.

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*.

Zhenhui Xu, Linyuan Gong, Guolin Ke, Di He, Shuxin Zheng, Liwei Wang, Jiang Bian, and Tie-Yan Liu. 2020. MC-BERT: Efficient language pre-training via a meta controller. *arXiv preprint arXiv:2006.05744*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 506–514.

Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, Luo Si, Yuan Ni, Guotong Xie, Zhifang Sui, Baobao Chang, Hui Zong, Zheng Yuan, Linfeng Li, Jun Yan, Hongying Zan, Kunli Zhang, Buzhou Tang, and Qingcai Chen. 2021a. CBLUE: A Chinese biomedical language understanding evaluation benchmark. *arXiv preprint arXiv:2106.08087*.

Ningyu Zhang, Qianghuai Jia, Kangping Yin, Liang Dong, Feng Gao, and Nengwei Hua. 2020. Conceptualized representation learning for Chinese biomedical text mining. *arXiv preprint arXiv:2008.10813*.

Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. 2021b. SMedBERT: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining.

In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5882–5893.

Hui Zong, Jinxuan Yang, Zeyu Zhang, Zuofeng Li, and Xiaoyan Zhang. 2021. Semantic categorization of Chinese eligibility criteria in clinical trials using machine learning methods. *BMC Medical Informatics and Decision Making*, 21(1):1–12.

A Pre-training Hyperparameters

We mostly use the same hyperparameters as ELECTRA (Clark et al., 2020) and do not conduct hyperparameter tuning during pre-training. As for those newly introduced hyperparameters, we sample $k = 5$ non-original tokens for a certain position in the MTS task, use a temperature $\tau = 0.07$ in the CSP task, and set the loss balancing tradeoffs $\lambda_1 = 50$, $\lambda_2 = 20$, $\lambda_3 = 1$. The full pre-training setting is listed in Table 7.

B Fine-tuning Hyperparameters

During fine-tuning, we mostly use the default setting as suggested by BERT (Devlin et al., 2019) and ELECTRA (Clark et al., 2020), listed in Table 8. We also use exponential moving average (EMA) with a decay coefficient α of 0.9999. Then for each task we specify a proper maximum sequence length, tune for each PLM the batch size, learning rate, and training epochs in their respective ranges, and determine optimal configurations according to dev performance. The full tuning ranges are listed in Table 9.

C CBLUE Benchmark

CBLUE (Zhang et al., 2021a)[10] is a benchmark for Chinese biomedical language understanding evaluation, consisting of 8 diversified biomedical NLP tasks as follows.

CMeEE: Chinese Medical Entity Extraction.[11] The task is to identify medical entities from a given sentence and classify the entities into nine categories including disease, symptom, drug, etc. The dataset contains 15K/5K/3K train/dev/test examples from textbooks of clinical pediatrics.

Hyperparameter	Value
Number of Layers	12
Hidden size	768
Intermediate size	3072
Number of attention heads	12
Attention head size	64
Embedding size	768
Generator size (multiplier for hidden size, intermediate size, number of attention heads)	1/3
Mask percentage	15
Learning rate decay	Linear
Warmup steps	10000
Learning rate	2e-4
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.999
Attention dropout	0.1
Dropout	0.1
Weight decay	0.01
Max sequence length	512
Batch size	384
Training steps	1.65M
Loss tradeoff λ_1	50
Loss tradeoff λ_2	20
Loss tradeoff λ_3	1
Multi-token selection k	5
Contrastive sequence prediction τ	0.07

Table 7: Pre-training hyperparameters.

Hyperparameter	Value
Learning rate decay	Linear
Warmup ratio	0.1
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999
Attention dropout	0.1
Dropout	0.1
Weight decay	0.01
EMA decay	0.9999

Table 8: Default fine-tuning hyperparameters.

CMeIE: Chinese Medical Information Extraction (Guan et al., 2020).[12] The task is to recognize both medical entities and their relationships from a given sentence according to a predefined schema. There are 44 relations defined in the schema, along with their subject/object entity types. The dataset contains about 14K/3.5K/4.5K train/dev/test examples, which are also from textbooks of clinical pediatrics.

CHIP-CDN: CHIP Clinical Diagnosis Normalization.[13] The task is to normalize original diagnostic terms into standard terminologies from the International Classification of Diseases (ICD-10), Beijing Clinical Edition v601. The dataset contains about

[10] https://github.com/CBLUEbenchmark/CBLUE

[11] http://www.cips-chip.org.cn/2020/eval1

[12] http://www.cips-chip.org.cn/2020/eval2

[13] http://www.cips-chip.org.cn/2020/eval3

Task	Batch size	Learning rate	Epochs	Length
CBLUE benchmark				
CMeEE	32	6e-5, 1e-4	2, 4, 8, 12	128
CMeIE	12	6e-5	50, 100, 150, 200, 250	300
CHIP-CDN	256	3e-5, 6e-5, 1e-4	2, 4, 8, 12, 16	32
CHIP-CTC	8, 16, 32	3e-5, 6e-5, 1e-4	2, 4, 8, 12, 16	160
CHIP-STS	8, 16, 32	3e-5, 6e-5, 1e-4	2, 4, 8, 12, 16	96
KUAKE-QIC	8, 16, 32	3e-5, 6e-5, 1e-4	2, 4, 8, 12, 16	128
KUAKE-QTR	8, 16, 32	3e-5, 6e-5, 1e-4	2, 4, 8, 12, 16	64
KUAKE-QQR	8, 16, 32	3e-5, 6e-5, 1e-4	2, 4, 8, 12, 16	64
Medical QA tasks				
cMedQNLI	8, 16, 32	3e-5, 6e-5, 1e-4	1, 2, 3, 4	512
webMedQA	16, 32, 64	1e-5, 2e-5, 3e-5	1, 2, 3, 4	512
NLPEC	32	2e-5, 3e-5, 6e-5	10, 20, 30, 40	512

Table 9: Hyperparameter tuning ranges on CBLUE and medical QA benchmarks.

6K/2K/10K train/dev/test examples collected from de-identified electronic medical records.

CHIP-CTC: CHIP Clinical Trial Classification (Zong et al., 2021).[14] The task is to categorize eligibility criteria of clinical trials into 44 predefined semantic classes including age, disease, symptom, etc. The dataset consists of about 23K/7.5K/10K train/dev/test examples collected from the website of Chinese Clinical Trial Registry.

CHIP-STS: CHIP Semantic Textual Similarity.[15] The task is to identify whether the semantics of two medical questions are identical or not. The dataset contains 16K/4K/10K train/dev/test question pairs collected from online healthcare services, covering 5 diseases including diabetes, hypertension, hepatitis, aids, and breast cancer.

KUAKE-QIC: KUAKE Query Intent Classification. The task is to classify the intent of a medical search query into one of 11 predefined categories like diagnosis, etiology analysis, medical advice, etc. The dataset contains about 7K/2K/2K queries in the train/dev/test split, collected from Alibaba QUAKE search engine.

KUAKE-QTR: KUAKE Query Title Relevance. The task aims to estimate the relevance between a search query and a webpage title. The relevance is divided into four levels: perfectly match, partially match, slightly match, and mismatch. The dataset contains about 24K/3K/5.5K query-title pairs in the train/dev/test split, collected from Alibaba QUAKE search engine.

KUAKE-QQR: KUAKE Query Query Relevance. Similar to KUAKE-QTR, the task is to estimate the relevance between two search queries Q_1 and Q_2. The relevance is divided into three levels: perfectly match, Q_2 is a subset of Q_1, Q_2 is a superset of Q_1 or mismatch. The dataset contains approximately 15K/1.6K/1.6K pairs of queries in the train/dev/test split. The queries are also collected from Alibaba QUAKE search engine.

D Medical QA Tasks

Besides CBLUE, we consider three medical question answering (QA) tasks, detailed as follows.

cMedQNLI: This is a Chinese medical QA dataset which formalizes QA as a question answer matching problem (Zhang et al., 2020).[16] Given a question answer pair, the task is to identify whether the answer addresses the question or not. The dataset contains about 81K/9K/10K question answer pairs in the train/dev/test split.

webMedQA: This dataset also formalizes medical QA as a question answer matching problem (He et al., 2019),[17] just like cMedQNLI. But it is much larger, containing roughly 250K/31.5K/31.5K question answer pairs in the train/dev/test split.

NLPEC: This is a multiple choice QA dataset constructed using simulated and real questions from the National Licensed Pharmacist Examination in China (Li et al., 2020).[18] Given a question along with five answer candidates, the task is to select the most plausible answer from the candidates using

[14] https://github.com/zonghui0228/chip2019task3

[15] http://www.cips-chip.org.cn:8000/evaluation

[16] https://github.com/alibaba-research/ChineseBLUE

[17] https://github.com/hejunqing/webMedQA

[18] http://112.74.48.115:8157

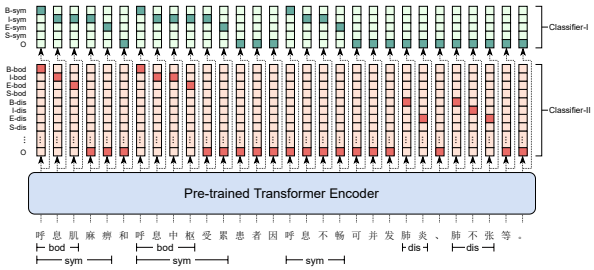

Figure 2: A running example illustrating the sequence tagging head used for CMeEE. Dark shaded entries represent a ground truth label of 1, and light shaded entries a ground truth label of 0.

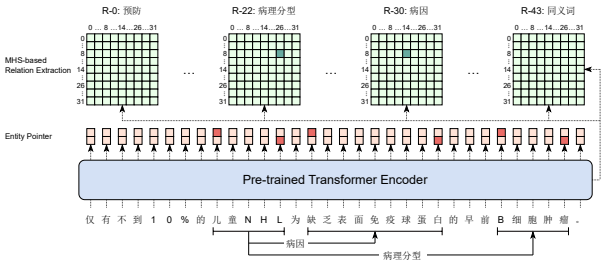

Figure 3: A running example illustrating the multi-head selection layer used for joint entity and relation extraction in CMeIE. Dark shaded entries represent a ground truth label of 1, and light shaded entries a ground truth label of 0.

textual evidences extracted from the official exam guide. The dataset contains about 18K/2.5K/0.5K questions in the train/dev/test split.

E Task-specific Heads

We devise lightweight task-specific heads on top of pre-trained Transformer encoders to solve downstream tasks in various forms. These task-specific heads are roughly categorized into five groups, used for named entity recognition, relation extraction, single sentence classification, sentence pair classification, and multiple choice QA, respectively.

Named Entity Recognition. CMeEE is the only task of this kind. It recognizes medical entities and classifies them into 9 predefined types. Nesting is allowed only in symptom entities, but not in entities of the other types. We therefore use a two-stream sequence tagging head for this task, one to identify symptom entities and the other to identify entities of the other 8 types. We choose the BIOES (*i.e.*, Begin, Inside, Outside, End, Single) tagging scheme (Ratinov and Roth, 2009). Given a sequence with its contextualized representations output by a pre-trained encoder, we build two classifiers on top of these representations. The first assigns each token in the sequence into 5 classes to annotate symptom entities (4 type-specific `B-`, `I-`, `E-`, `S-` tags plus `O` tag), while the second assigns it into 33 classes to annotate entities of other types (32 type-specific `B-`, `I-`, `E-`, `S-` tags plus `O` tag). The two classifiers are trained jointly with a 1:1 balanced combined loss. Figure 2 gives a running example illustrating this two-stream sequence tagging head.

Relation Extraction. CMeIE is the only task of this kind. It extracts subject-relation-object triples according to a predefined schema. There are totally 44 relations defined in the schema and overlapping is allowed between these relations, *i.e.*, one entity may belong to multiple triples of different relations (Zeng et al., 2018). To solve this overlapping problem, we use a multi-head selection (MHS) layer for joint entity and relation extraction (Bekoulis et al., 2018). As illustrated in Figure 3, an entity pointer is adopted to identify start and end of entity spans, and then an MHS mechanism is further employed to recognize possible relationships between pairs of entity spans. The MHS module predicts if there exists a relation k between a subject entity starting at position i and an object entity starting at position j for every i, j, and k. This prediction probability is calculated via a relation-specific biaffine operation imposed upon the starting token representations of subject and object entities. Finally, we jointly train the entity pointer and MHS-based relation extractor via a combined loss with balancing ratio of 1:50.

Single Sentence Classification. CHIP-CTC and KUAKE-QIC are tasks of this kind, which classifies a given sentence into one of a set of predefined categories. We simply build a softmax classifier on top of the final representation corresponding to the initial `[CLS]` token for this classification task.

Sentence Pair Classification. The sentence pair matching tasks of CHIP-STS, KUAKE-QTR, and KUAKE-QQR, as well as the medical QA tasks of cMedQNLI and webMedQA are of this kind, aiming at predicting the semantic relationship between a pair of sentences according to a set of predefined labels. CHIP-CDN, after normalized terms have been retrieved for each original term, can also be formalized as a task of this kind, the aim of which is to judge if a normalized term matches the original term or not. Given a pair of sentences (S_1, S_2), we pack them into a single input sequence "`[CLS]` S_1 `[SEP]` S_2 `[SEP]`", and feed this sequence into a

pre-trained encoder. Then we build a softmax classifier on top of [CLS] representation to conduct sentence pair classification. For CHIP-CDN, we retrieve 100 candidate normalized terms for each original term from the whole ICD-10 vocabulary using Elasticsearch before pairwise classification.

Multiple Choice QA. NLPEC is the only task of this kind. It selects the most plausible answer from 5 answer candidates for a given question. Textual evidences are also provided along with the question. Let Q denote the question, $\{A_1, A_2, A_3, A_4, A_5\}$ the answer candidates, and T the textual evidence. For each answer candidate A_i, we pack it with the question Q and textual evidence T, and construct a single input sequence "[CLS] A_i [SEP] Q [SEP] T [SEP]". We feed this sequence into a pre-trained encoder, and use [CLS] representation to estimate if A_i answers Q given textual evidence T. In this fashion, we transform multiple choice into binary classification. At inference time, the candidate with highest probability is chosen as the correct answer.