# StaResGRU-CNN with CMedLMs: A Stacked Residual GRU-CNN with Pre-trained Biomedical Language Models for Predictive Intelligence

Pin Ni[1], Gangmin Li[2], Patrick C. K. Hung[3], Victor Chang[4,5]

## Abstract

As a task that requires strong professional experience as support, predictive biomedical intelligence cannot be separated from the support of a large amount of external domain knowledge. By using transfer learning to obtain sufficient prior experience from massive biomedical text data, it is essential to promote the performance of specific downstream predictive and decision-making task models. This is an efficient and convenient method, but it has not been fully developed for Chinese Natural Language Processing (NLP) in the biomedical field. This study proposes a Stacked Residual Gated Recurrent Unit-Convolutional Neural Networks (StaResGRU-CNN) combined with the pre-trained language models (PLMs) for predictive tasks based on biomedical texts. By exploring related paradigms in biomedical NLP based on external expert knowledge transfer learning and comparing some Chinese and English language models. We found some key points that have not been developed or have practical applicability difficulties in the Chinese biomedicine field. Therefore, we also propose a Chinese bioMedical Language Model series (CMedLMs) with a detailed downstream tasks evaluation. By using transfer learning, language models are introduced with prior knowledge to improve the performance of downstream tasks and solve

[1] School of Engineering, University College London, London, UK
[2] School of Computer Science and Technology, University of Bedfordshire, Luton, UK
[3] Faculty of Business and Information Technology at Ontario Tech University, Canada
[4] Artificial Intelligence and Information Systems Research Group, School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, UK
[5] Corresponding Author

specific predictive NLP tasks related to the Chinese biomedical field to serve the predictive medical system better. Additionally, a free-form text Electronic Medical Record (EMR)-based Disease Diagnosis Prediction task is proposed, which is used in the evaluation of the analyzed language models together with Clinical Named Entity Recognition, Biomedical Text Classification tasks. Our experiments prove that the introduction of biomedical knowledge in the analyzed models significantly improves their performance in the predictive biomedical NLP tasks with different granularity. And our proposed model also achieved competitive performance in these predictive intelligence tasks.

*Keywords:* Natural Language Processing, Predictive Intelligence, Biomedical Text Mining, Named Entity Recognition, Text Classification, Transfer Learning, Pre-trained Language Model

## 1. Introduction

About one-fifth of the world's population speaks Chinese. Hundreds of millions of Chinese speakers' medical information is contained in large and complex electronic medical data management systems. At present, in the situation of such a huge number of electronic medical data, it is imperative to implement convenient and effective means of processing these massive amounts of textual data. The processing part means using deep learning-based NLP, modeling complex medical text data through deep learning-based NLP to build a more robust medical prediction system to provide more accurate clinical suggestions. Therefore, it can become another new and effective auxiliary diagnosis method besides relying on expert experience.

Some of the most effective paradigms today are based on the newly emerging pre-trained language models. They were introduced with a large amount of prior knowledge to raise the performance of downstream NLP tasks (e.g., EMR-based healthcare decision reference). These new methods can compensate for the predicament of insufficient training data in supervised learning and enhance the ability to encode the semantics of context in texts adequately. The effectiveness of these models that use prior knowledge to solve different tasks they have been demonstrated in many recent experiments on various mainstream NLP tasks [1, 2]. It is now widely recognized by professionals as a new frontier in the NLP field. However, pre-trained language models (trained by large-scale prior knowledge) and their downstream task models, specifically in the Chinese biomedical field, have not been studied yet. Consequently, the NLP tasks processing Chinese biomedical data are often narrowed by the lack of prior domain knowledge and training resources. Compared with similar tasks using other languages (e.g., English), it is often not possible to reach a comparable performance in practical business scenarios, thus limiting the deployment in real-world medical Human-Computer Interaction (HCI) scenarios.

Many studies confirm that most of the selection of downstream task models have a limited impact on the performance of specific tasks [3, 4, 5, 6, 7]. At the same time, several studies [8, 9, 10] have begun to explore the introduction of external knowledge in language representation models. However, due to the late start in the usage of Chinese E-health technologies and the privacy problems related to storing and processing medical data. These all have played a certain role in limiting the development of Chinese biomedical language models. In addition, the introduction of Chinese biomedical field knowledge into pre-trained models poses a number of challenges, the first being that trainable resources, such as medical data, are difficult to access, and knowledge in specific professional fields is often abstract and diverse, making it difficult to comprehensively and deeply cover. Despite the terminology specifications, the expression of medical knowledge is still complicated and diverse in actual business (e.g., abbreviations, polysemy, free-form writing). Entities may have completely different definitions in different contexts. This, combined with frequent mixtures of Chinese and English terminologies, increases the difficulty for the model to process complete sentences correctly. Finally, the inaccuracy of Chinese word segmentation often leads to unsatisfactory actual results.

For the reasons detailed above, NLP in the Chinese biomedical field is currently incomparable to its English counterpart. There is a large amount of open medical text data, corpus (JNLPBA, NCBI, BC2GM, etc.) and related research competitions (e.g., i2b2, SemEval, TREC Medical/CDS) [11] that deal with NLP tasks in English in the biomedical field. For example, the 2020 and 2019 studies [1, 12] on biomedical pre-trained language models were the pioneers in the English field. And the evaluation of downstream tasks [13, 14] has recently begun to be adopted, proving the value of this emerging research field. Pre-trained language models, which have absorbed a lot of the Chinese biomedical domain knowledge, can benefit the related Chinese NLP tasks. A Chinese medical record can be served as the context representation of medical terms and be used to train a model to enhance its capability in executing downstream prediction-related NLP tasks. This also provides strong support for the construction of medical predictive intelligent systems.

In this study, we designed a Stacked Residual Gated Recurrent Unit Convolutional Neural Networks (StaResGRU-CNN) combined with the pre-trained SOTA language models for prediction tasks based on biomedical texts and achieved competitive performance in these tasks. We also introduce CMed-Language Models (CMedLMs), a series of domain-specific, pre-trained language models for downstream tasks with Chinese biomedical textual data. The series is made up of 3 major pre-trained language models (BERT, Word2Vec, GloVe) trained with a large, real Chinese biomedical corpus. We use them (CMed-BERT, CMed-Word2Vec, and CMed-Glove) with their downstream models (including our proposed) to perform three text-based prediction tasks with different granularity (Clinical Named Entity Recognition [a term fragment boundary and category prediction task], Biomedical Text Classification [label prediction task], and Free-form Text EMR-based Disease Diagnosis Prediction) in detailed and extensive comparative experiments (including ablation experiments). In addition, we also discuss some practical difficulties with EMR-based clinical diagnoses. This research is also currently the first comprehensive work of pre-trained biomedical models with extensive experimental evaluations to the best of our knowledge. This contributes to medical predictive intelligence tasks related to biomedical NLP.

## 2. Related Works

### *2.1. Predictive Intelligence*

Predictive intelligence is very good at dealing with known knowns and can fill the gap between know unknowns well. This is usually done through a large amount of historical data or prior knowledge as the training samples so that the model can learn the information carried in these massive data to solve specific predictive tasks [15, 16, 17, 18, 19]. These predictive tasks also play many roles in the field of natural language processing [20, 21, 22, 23, 24]. Most predictive NLP tasks can be transformed into discrete data-oriented classification tasks. Therefore, the label prediction of text sequences at different granularities can be clearly defined as multi-class classification, multi-label classification, or sequence labeling tasks (e.g., Named Entity Recognition, a term fragment boundary and category prediction task).

And the machine cannot predict unprecedented things. Therefore, in many cases, the machine has a high probability of making a misjudgment in the prediction of unknown data. And through large-scale transfer learning (e.g., pre-training language model) or incremental learning of known knowledge from external sources, it ameliorates the weaknesses of the correct results inferred by the wrong decision-making process or similar situations. This can also enhance the decision-making accuracy and robustness of predictive intelligence to a certain extent.

### *2.2. Deep Learning-based NLP*

Deep learning technologies derived from neural network models are the most researched machine learning entities in the contemporary academic field. Many studies proved the effectiveness of these models and led to a change of views in the entire researcher community. Among them, the most representative of the deep learning models, including Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), are commonly used as models in developing natural language processing strategies. These are also widely used in many major NLP tasks: sequential tagging, classification, text generation, etc. Studies confirmed the effectiveness of deep learning paradigms for sequence labeling tasks. Tomori et al. [25] proposed and trained a DNN+R model (a method that refers to real-world data to improve Named Entity Recognition (NER) specific to a domain.) and found that it performed much better on NER tasks than other simpler DNN models. Lample et al. [26] proposed two neural network architectures: a bidirectional LSTMs, with Conditional Random Fields (CRF), while another network constructs and segments labels, obtaining the best NER performance results ever reported in standard evaluation settings, even if compared to models that make use of use external resources, such as gazetteers. Bharadwaj et al. [27] added a layer of phoneme features to Lample et al.'s LSTM and achieved an even better performance in a monolingual setting using supervision.

Deep learning also challenges traditional methods in classification tasks. Venkataraman et al.[28] used LSTM-RNNs classifying unstructured medical descriptions, reaching accuracy and F1 score higher than those of decision trees [29] and random forests [30]. Mironczuk et al. [31] quantitatively analyzed the literature on text classification in springer, Elsevier, ACM, and IEEE repositories, studied what the most impacting elements in the performance of text classification tasks are. They found that many works relied on the generation of embeddings to provide richer semantic representations for classifiers [32, 33, 34, 35]. The authors also conducted a detailed literature review of machine learning type text classification methods, including neural networks. At the same time, deep learning model strategies are also constantly being updated iteratively. Du et al. [36] proposed recurrent BLS (R-BLS) and gated BLS (G-BLS), two novel text classification learning methods derived from a flat neural network known as Broad Learning System (BLS). These two methods can simultaneously learn from two sets of inputs, making them more accurate than LSTM. Thanks to the noniterative learning of BLS, the training process is faster than that of LSTM. Kim et al. [37] used a capsule network architecture for text classification tasks, proving how it provides more advantages than CNNs. At the same time, it also proved that its accuracy is better than that of SA-LSTM [38], VA LSTM [39] and DCNN [40] on seven benchmark datasets.

### *2.3. Pre-trained Language Models*

Pre-trained Language Models have been proven to improve the performance of NLP models based on deep learning [38, 41, 42, 43] on different benchmark datasets. Traditional vector representations with varying granularities (e.g., Word2Vec [44] and GloVe [45]) tend to be uncontextualized and encapsulate all meaning within a single vector. However, some of the latest and most advanced models, like ELMo [41] and BERT [46], consider the context in which they operate and can consequently achieve better performance.

BERT [46] is a language model designed to pre-trained deep bidirectional representations from the unannotated text by collectively turning both the left and the right context in every layer. As a result, a pre-trained BERT model can be fine-tuned using only one additional output layer to create state-of-the-art (SOTA) models for a multitude of tasks, such as question-and-answering, language inference and named entity recognition, without any substantial task-specific architecture modifications [47].

## 3. Methodology

This research proposes the Stacked Residual Gated Recurrent Unit-Convolutional Neural Networks (StaResGRU-CNN). This method is inspired by the model structure of Recurrent Convolutional Neural Networks (RCNN) [48]. Compared with the previous method, this method has some improvements in multiple tasks. It can be divided into three aspects:

- Compared with Recurrent Neural Networks (RNN) as a model for text context information modeling, Gated Recurrent Unit (GRU) can handle long-distance dependencies better. Furthermore, compared with RNN and Long Short Term Memory (LSTM), it has a more simplified structure and fewer parameters, so the computing efficiency of each layer has a certain improvement.

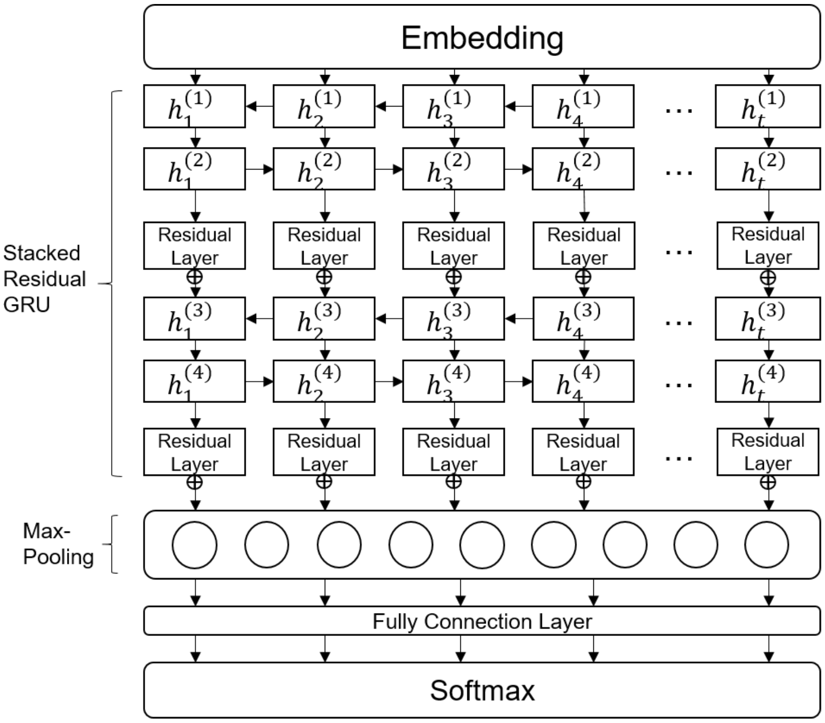

Figure 1: StaResGRU-CNN Structure

- This method uses Stacked GRU as the basis of feature modeling to represent features by constructing deeper networks.

- The Residual Connection is also introduced in Stacked GRU to center the layer gradients and the propagation error.

This model inherits the structural characteristics of RCNN: it makes full use of the advantages of RNN and CNN. Similar to the structure of RCNN, this model mainly creates a four-layer Stacked GRU (Dual Bidirectional GRU) with Residual Connection and combines Max-Pooling in CNN. The model can determine which features are critical in tasks related to classification to capture the key components in the text. Its specific structure is shown in Fig. 1.

The specific process can be divided into the following five steps:

- Use Stacked GRU to obtain contextual information.

- Add a Residual Layer after each BiGRU to speed up the convergence rate.
- Mapping the vectors to lower dimensions.
- At each position in the hidden size vector, take the maximum value of the all-time series to obtain the final feature vector (Max-Pool).
- Softmax classification.

Therefore, the proposed model can be expressed in the following parts:

### *3.1. Stacked Residual GRU*

As a variant of RNN, GRU uses the current input $x_t$ and the hidden state $h_{t-1}$ passed down from the previous node. This hidden state contains information about the previous node. By combining $x_t$ and $h_{t-1}$, GRU will get the output of the current hidden node $y_t$ and the hidden state$h_t$ passed to the next node. In the GRU, the gate state is obtained through the state $h_{t-1}$ transmitted from the previous node and the input $x_t$ of the current node. Among them, $r$ represents the reset gate, and $z$ represents the update gate. $\sigma$ is the sigmoid function, through which the data can be transformed into a value in the range of $[0, 1]$ to act as a gate signal.

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \tag{1}$$

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \tag{2}$$

After obtaining the gate signal, first to use the reset gate to get the data $r_t \cdot h_{t-1}$ after "reset", then concatenate $h_{t-1}$ with the input $x_t$, and then scale the data to the range of $[-1, 1]$ through $tanh$ activation function, that is:

$$\tilde{h}_t = \tanh\left(W \cdot [r_t \cdot h_{t-1}, x_t]\right) \tag{3}$$

The $h^{'}$ here mainly contains the current input $x^t$ data and specifically adds $h^{'}$ to the current hidden state, which can also be considered as "Memorized the state of the present moment."

Another key process is to "renew" the memory. This stage includes two sub-steps: forgetting and remembering. The updated expression is as follows:

$$h_t = (1 - z) \odot h_{t-1} + z \odot h' \tag{4}$$

This step is to forget the information of some dimensions in the passed $h_{t-1}$, and add the information of some dimensions input by the current node.

The gate signal $z$ has a range of $[0, 1]$. The closer to 1, the more data is "remembered", and the closer to 0 the more "forgotten". $(1 - z) \odot h_{t-1}$ represents the selective "forgetting" of the original hidden state. $z \odot h^{'}$ means to selectively "memorize" the $h^{'}$ containing the current node information.

In our model, we stack multiple layers of bidirectional GRU (BiGRU) together, where the hidden representation $h_t^{(l)}$ of the previous layer is used as the input of the next layer, and $l$ is the layer. Therefore, the hidden state of time $t$ in the $l$ layer can be expressed as:

$$\overleftarrow{h_t^{(l)}} = \overleftarrow{GRU}\left(x_t, \overleftarrow{h_{t-1}^{(l)}}\right) + x_t \tag{5}$$

$$\overrightarrow{h_t^{(l)}} = \overrightarrow{GRU}\left(x_t, \overrightarrow{h_{t-1}^{(l)}}\right) + x_t \tag{6}$$

### *3.1.1. Residual Connection Layer*

When multiple layers of neurons are stacked, the neural network will degenerate due to the low convergence speed of training errors. However, the residual network can solve this problem. Therefore, inspired by ResNet [49], we add Residual Connection Layer at the end of each BiGRU. This method accelerates convergence by transferring residual information. Inspired by Toderici et al. /citetoderici2017full, $h_t^{(o)}$ represents output, and $W$ and $U$ represent convolutional linear transformation. i.e., they are composites of Toeplitz matrices with padding and stride transformations.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tanh\left(W x_t + U\left(r_t \odot h_{t-1}\right)\right) + \alpha_h W_h h_{t-1} \tag{7}$$

$$h_t^o = h_t + \alpha_x W_{ox} x_t \tag{8}$$

In addition, each Residual Connection Layer will not add any additional parameters that need to be learned, so it will not increase the complexity of the model.

### *3.2. Max-pooling, Fully Connection Layer and Softmax*

#### *3.2.1. Max-pooling*

The overall structure of the StaResGRU-CNN mainly uses Stacked GRU to replace the convolutional layer in CNN to obtain the semantic representation of the context and combines the Max-pooling layer in CNN. Through this layer, a fixed-length vector can be obtained. And obtain the latent semantic information that best represents the meaning of the text.

$$y^{(2)} = \max_{i=1}^{n} y_i^{(1)} \tag{9}$$

#### *3.2.2. Fully Connection Layer*

$$y^{(3)} = W y^{(2)} + b \tag{10}$$

#### *3.2.3. Softmax*

Finally classify through Softmax.

$$p_i = \frac{\exp\left(y_i^{(3)}\right)}{\sum_{k=1}^{n} \exp\left(y_k^{(3)}\right)} \tag{11}$$

## 4. Tasks Description

### *4.1. Pre-trained Language Model*

The combination of various language models to enhance the effect in NLP downstream tasks has proven to be effective [50]. Researchers have lately been applying prior external knowledge in specific fields to run downstream tasks, effectively improving the adaptability and accuracy of these results. Since the current research development in the field of Chinese biomedical language models is almost none, the research also focuses on introducing Chinese biomedical knowledge into language representation models. Through this potential paradigm, it will attempt to improve downstream tasks related to Chinese biomedical NLP. Therefore, inspired by the idea in [50], we divide the pre-trained language model into two categories according to the characterization type: Word Embedding (Word2Vec, GloVe) and Seq2Seq (BERT).

### *4.2. Fine-Tuning the Pre-trained Language Models*

A language representation model is greatly influenced by the scale and quality of the training data in downstream training tasks. Most of the existing pre-trained language models use commonly available data, such as online encyclopedias, Q&A discussion groups, and forums as sources [51, 46, 52]. For another, biomedical texts contain a large number of specific expressions (such as “Radical thyroidectomy for thyroid carcinoma”) and proper nouns (e.g., “Electronic Medical Record Basic Dataset Specification of P.R.China”), which are understandable by professional medical practitioners only. Consequently, language models that are pre-trained using wide-domain sources have more limited performance in specific downstream tasks belonging to the biomedical field. Especially important when dealing with Chinese biomedical texts, it is necessary to consider the number of training samples available, the word segmentation accuracy, the vocabulary, and the expressions’ diversity, as well as the coding uniformity of the characters (e.g., double-byte, single-byte, special symbols, GBK encoding), the expertise level of the authors, the uniformity of the text format specification and the quality of the dataset (noise content ratio), among other constraints. On a semantic level, understanding the meaning of sentences is more difficult in Chinese than in English [53]. Therefore, a pre-trained model, trained with specialized medical data in Chinese, will be used for solving three main biomedical NLP fine-grained prediction-related tasks (Clinical NER, Biomedical Text

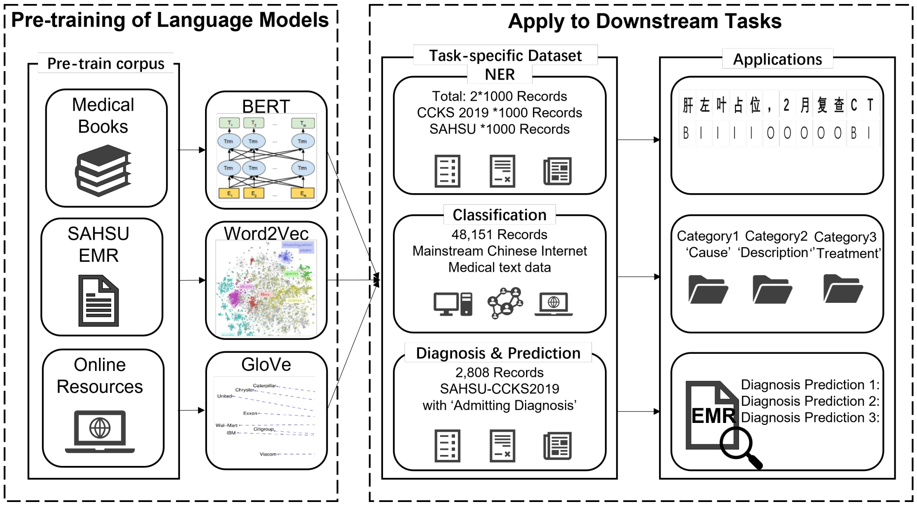

Figure 2: Overview of the overall structure of Chinese Medical Language Models (CMedLMs)

Classification, and Free-form Text EMR-based Disease Diagnosis Prediction). Both crowdsourced and actual Chinese EMR datasets are used in the experiments to evaluate the performance of the pre-trained language models on the various downstream tasks (Fig. 2).

### *4.2.1. Clinical Named Entity Recognition (Clinical NER)*

Clinical NER, an essential task in biomedical text mining, includes the identification of a large amount of domain-specific nouns and can be transformed into NLP extraction and classification tasks [54, 55]. Specifically, in the biomedical field, its task can be described as: "For a given set of EMR text documents, identify and extract all medical nouns, then classify them into one of some predefined categories (e.g., diseases, treatments, examinations)." Many researchers are currently focusing on a type of pipeline that combines LSTM and CRF. Others focus on this type of incremental pipeline [56, 57] instead. And other SOTA methods adopt pre-trained language models to obtain words or even characters representations [58, 59, 60]. However, in these models, biomedical-specific knowledge is not usually included as prior knowledge, thus limiting their effectiveness. Meanwhile, this task can be regarded as a fine-grained predictive task at the next level to a certain extent. This prediction task aims to enable the model to recognize the boundaries of term fragments in the text sequence and which category the fragments within a certain byte range belong to.

### *4.2.2. Biomedical Text Classification*

Biomedical Text Classification is one of the main tasks of Biomedical NLP. Its goal is to infer a label (or a collection of labels) for a given text (be it a sentence, a document, etc.). Its role in the biomedical field is to label texts based on their specific area of interest. Classification is an indispensable core step in information retrieval, leading to automatic sorting of electronic medical records, hospital outpatient guiding robots, doctor-patient dialogue intention identification, Disease Diagnosis Prediction and more. Moreover, it is also a fine-grained category prediction task based on prior knowledge/expert experience. Therefore, as one of the most critical downstream tasks in the NLP field [61, 62, 11], classification can be used as a task to evaluate the performance of pre-trained models. We use biomedical data captured from the four mainstream Chinese online medical knowledge encyclopedias (Tab. 2) in order to test the performance of the multi-class classification task. We will finally compare these results with those of the original models we used have used before applying any changes.

### *4.2.3. Free-form Text EMR-based Disease Diagnosis Prediction*

Outpatient diagnostic records contain a detailed history of the progression and treatment of a patient's illness. When visiting outpatients, doctors make some initial judgments on likely diseases based on patients' chief complaints, past medical history, and medical conditions. This information is of great help in possible later stages to accelerate diagnosis processes.

Free-form Text EMR-based Disease Diagnosis Prediction is a contextualized action derived from the classification task. However, this task has higher requirements than ordinary classification tasks: for example, it has a larger number of disease diagnosis labels (Super Multi-class Classification). There are also many cases where a single medical record can diagnose multiple diseases (Multi-label Classification). In our study, the diagnosis of a patient with an illness is inferred from two pieces of data: a similar diagnosis found in the EMR dataset and information specific to the patient, such as the content of the patient's complaint, the history of the illness, on-site checkups, etc.

By training the model on the contents of EMRs and using preliminary diagnostic information (labels), the task can be transformed into super multi-class and multi-label classification (each disease diagnosis being a label) with EMR text mining. Therefore, we model the HPI (History of Present Illness) records in the EMRs and use the information from the corresponding "initial diagnosis", or "admitting diagnosis", as label sets to train and test the model and make diagnostic predictions for a given disease record; then, using multiple deep learning models, we analyzed the contents of a medical record to predict the most likely disease given the symptoms. This will benefit the medical community as a way to provide doctors with a predictive reference for diagnosis through expert experience (a large amount of potential knowledge stored in biomedical text data, including electronic medical records, medical books, encyclopedias, etc.).

## 5. Experiments

### *5.1. Experimental Environment*

The machine where we deployed our CMed-BERT pre-trained model a TPU v3-8 (128 GiB VRAM), with 4 vCPU and 15 GB of RAM. The CMed-Word2Vec and CMed-GloVe models and their downstream tasks are trained with dual NVIDIA 1080Ti GPUs (with 11 GB VRAM for each), an Intel Xeon CPU E5-2678 v3 and 64GB of RAM.

### *5.2. Description of Pre-train Corpora and Downstream Tasks' Datasets*

#### *5.2.1. Description of Pre-train Corpora*

We have collected a large amount of Chinese biomedical literature and real-world electronic medical record data (both in Chinese and English) for pre-training our language models. These are mainly taken from 3 sources: Chinese online biomedical encyclopedias, EMR datasets provided by the Second Affiliated Hospital of Soochow University (SAHSU, the large version), and Chinese biomedical books. The specific quantities, categories, and other relevant information can be found in Tab. 1.

Table 1: Description of Pre-trained Corpora

| Corpus Source | Corpus Description | Size (Chinese Characters) |
|---|---|---|
| Medical Books | We used 13 books as a corpus, including *Manual For ICU Attending Doctor*, *Reading X-ray Guide*, *CT Diagnostics*, *Immunology*, *Pathology*, *Clinical Drug Therapy*, *Psychiatry*, *Clinical Electrocardiogram Detailed Analysis and Diagnosis*, *Tumor*, *Surgery*, *Hyperemia*, *Gynaecology* and other books of disciplines. | 4,384,503 (4M) |
| SAHSU[5] | The electronic medical records of the Second Affiliated Hospital of Soochow University, including 5,090 electronic medical records from the 3 departments of General Surgery, Intervention and Oncology from the last 2 years. | 2,002,202 (2M) |
| Online Resources | From the 4 mainstream Chinese websites in the medical field named “39 Health”[1], “XunYiWenYao”[2], “Feihua Health”[3], “NetEase Health”[4] to capture the text data about medical encyclopedias such as disease symptoms, drugs, medical term explanations, medical cases, treatment plans, etc. | 29,092,216 (29M) |

[1] www.39.net, [2] www.xywy.com, [3] www.fh21.com.cn, [4] jiankang.163.com, [5] The large version of SAHSU electronic medical records dataset, separated from the full version of the SAHSU dataset.

### 5.2.2. Description of Downstream Tasks' Datasets

We have selected specific datasets for different tasks, which are all mainly composed of EMRs and online open medical text data. Details are listed in Tab. 2.

Table 2: Description of Downstream Tasks' Datasets

| Task Name | Dataset Description | Size (Records) |
|---|---|---|
| Clinical Named Entity Recognition | 1,000 randomly selected labelled Electronic Medical Records from CCKS 2019 and SAHSU[1] respectively for testing Clinical Named Entity Recognition task | 2 × 1,000 records |
| Biomedical Classification | The dataset is taken from 4 mainstream Chinese online medical encyclopedias and Q&A websites (Refer to the “Online Resources” section of the table above), for a total of 48,151 records. These typically include 6 sections: “Cause”, “Description”, “Diagnosis”, “Prevention”, “Symptom”, “Treatment”. These sections are used as the basis for multi-class classification. | 48,151 records |
| Free-form Text EMR-based Disease Diagnosis & Prediction | An EMR-based Disease Diagnosis Prediction dataset with “Admitting Diagnosis” labels integrated by SAHSU[1] and CCKS 2019 after data cleaning. (Named SAHSU-CCKS) | 2,808 records |

[1] The small version of SAHSU electronic medical records dataset, separated from the full version of the SAHSU dataset

### *5.3. Pre-training Seq2Seq Language Models Process*

BERT is a huge Seq2Seq type language model; Its training process is longer and more complicated than that of Word Embedding type models. When pre-training CMed-BERT, due to the huge memory consumption of the BERT and the limited memory of the TPU (TPU v3-8 with 128 GB of memory), the training program can only be run when the batch size is adjusted to 32 and the maximum sequence length is adjusted to 384. Pre-training included 120k steps and took 12 days, 20 hours, 11 minutes 10 seconds, with a final loss of 1.633, which is already a relatively ideal result.

The hyper-parameter settings for pre-training CMedLMs (CMed-BERT, CMed-Word2Vec and CMed-GloVe) and different types of downstream models can be found in Tab. 3-6.

In addition, the hyperparameter settings of our proposed model depend on the tasks involved and the datasets used. Therefore, we refer to and adopt the hyperparameter setting strategies commonly used in previous studies to set specific parameters for different tasks and datasets. The general parameter settings used in the different datasets of the three tasks in the experiment include the batch size is 64, epochs are 100, dropout is 0.5, the activation function in CNN is ReLu, and the final activation layer of the overall model using Softmax.

Table 3: List of Hyper-parameters Settings and Training Process Data Records for Pre-training CMed-BERT

| | |
|---|---|
| **max_predictions_per_seq** | 77 |
| **max_sequence_length** | 384 |
| **learning_rate** | 1e-4 |
| **num_warmup_steps** | 10,000 |
| **batch_size** | 32 |
| **smoothed** | 1.625 |
| **num_train_steps** | 120.00k |
| **time-consuming** | 12d 20h 11m 10s |
| **final loss** | 1.633 |

Table 4: List of Hyper-parameter Settings for Pre-training CMed-Word2Vec[1]

| | | | |
|---|---|---|---|
| **sentences** | None | **workers** | 3 |
| **size** | 100 | **min_alpha** | 0.0001 |
| **alpha** | 0.025 | **sg (skip-gram)** | 0 (disable, adopt CBOW) |
| **window** | 5 | **negative** | 5 (negative sampling,<br>5 noise words) |
| **min_count** | 5 | **cbow_mean** | 1 (enable) |
| **max_vocab_size** | None | **hashfxn** | hash |
| **sample** | 1e-3 | **iter** | 5 |
| **seed** | 1 | **null_word** | 0 |
| **trim_rule** | None (min_count) | **sorted_vocab** | 1 |
| **batch_words** | MAX_WORDS_IN_BATCH | **hs (hierarchica softmax)** | 0 (negative sampling) |

[1] The overall hyper-parameter setting of CMed-GloVe is consistent with CMed-Word2Vec.

Table 5: List of Hyper-parameter Settings for the Overall Models and LSTM/GRU Components

| | |
|---|---|
| **batch_size** | 64 |
| **epochs** | 100 |
| **dropout** | 0.4 |
| **bilstm_units** | 512 |
| **dense_units** | 512 |
| **dense layers activation** | Softmax |

Table 6: List of Hyper-parameter Settings in the CNNs Component

| | |
|---|---|
| **filters** | 64 |
| **kernel_size** | 3 |
| **padding** | same |
| **activation** | ReLU |

To compare the effects of different semantic segmentation (Chinese word segmentation) applied to biomedical texts, we designed two types of segmentation methods to operate on raw corpus: word-level and character-level segmentation. In CMed-Word2Vec and CMed-GloVe, a corpus of nearly 36 million characters was mapped to both word vectors and character vectors for training purposes, after going through the processes of word segmentation and character

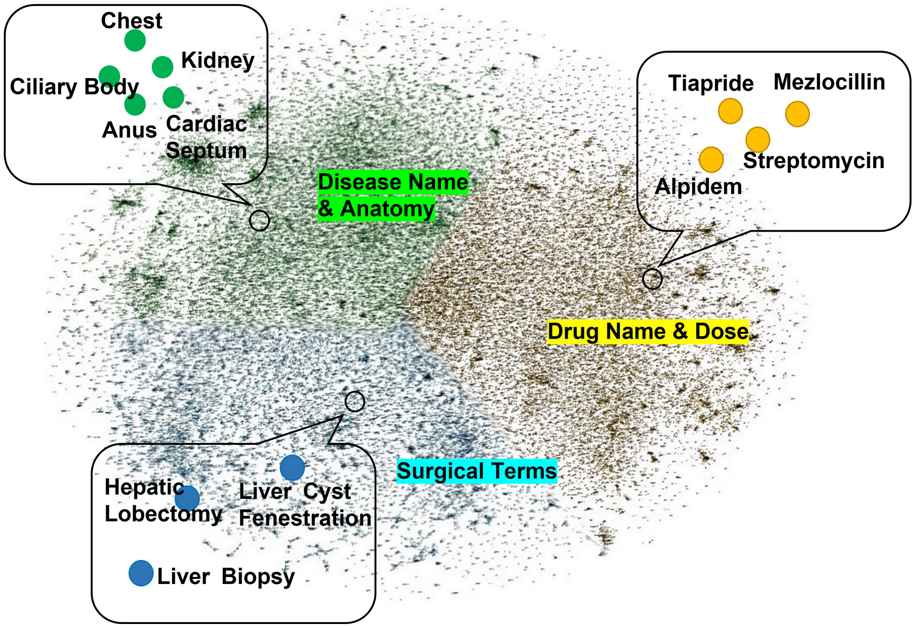

Figure 3: 2-D scatter graphs of CMed-Word2Vec (Word-Level)

segmentation, and finally generates a 50-dimensional vector for each word or character. Consequently, we obtained a total of 4-word embedding models: 2 CMed-Word2Vec models (Word-Level and Character-Level) and 2 CMed-GloVe models (Word-Level and Character-Level). Fig. 3 shows 2-D scatter graphs of the word vectors (CMed-Word2Vec). Taking Fig. 3 as an example (approximate range), the drug name and dose are mainly embedded in the yellow region. The anatomy term and disease name are mainly concentrated in the green region; the blue area mostly contains the vocabulary of surgical terms. These word embeddings are trained from a large-scale biomedical corpus containing mixed Chinese and English.

Additionally, we conduct detailed ablation studies to verify the actual performance of the following three types of language models on different downstream tasks: bare embedding, the original pre-trained language models (based on general knowledge pre-training), the Chinese biomedical models (based on Chinese biomedical domain knowledge pre-training). In these models, we use GloVe pre-trained based on the official offline version of Wikipedia (Chinese) [63] and Word2Vec pre-trained on Chinese Wikipedia and Baidu encyclopedia[6] [64] as the word embedding for ablation experiments. At the same time, we also use "BERT-Base, Chinese" [46] provided by Google as the embedding of the original BERT model. For the bare embedding method in the ablation experiments, a random parameter initialization method is used to generate each token vector.

## 6. Results and Analysis

### *6.1. Clinical Named Entity Recognition*

CCKS 2019 Task1 is an academic evaluation task for Chinese EMR Named Entity Recognition. The dataset published together with the task is also the largest and only open Chinese EMR medical entity recognition dataset globally. Its records are organized in 6 categories: "Drugs", "Anatomical Sites", "Diseases and Diagnoses", "Surgery", "Laboratory Inspection", and "Image Examinations". We randomly selected 1,000 complete samples and divided them into 3 datasets for training, validation, and testing with an 8:1:1 random split before training. In addition, we have also used the EMRs provided by the Second Affiliated Hospital of Soochow University, randomly selecting 1,000 sample records and splitting them into the same categories as above. To uniform the two datasets, we manually labeled the latter records to give each of the six categories of the dataset from CCKS 2019 Task1. The annotation specification is also consistent with the details in CCKS 2019 Task1.

### *6.1.1. CCKS 2019*

On the CCKS 2019 dataset, the performance of CMed-BERT compared to vanilla BERT on the five downstream models has an average of F1-Score improvement of 2.04% per model. The average of F1-Score of each downstream model reached 73.31%, whereas the official BERT has 71.27% and fine-tuned

[6] baike.baidu.com

BERT 72.56%. In addition, compared to word embedding models like general GloVe, Word2Vec, and Bare Embedding, CMed-BERT performs better, with a higher average of F1-score of 18.98%, 16.82%, and 13.35%, respectively. Among these models, the best-performing one follows a CMed-BERT-StaResGRU-CNN-CRF pipeline; its F1-Score reached 76.32%, 1.33% higher than with a BERT-ResGRU-CNN-CRF pipeline (the best performing pipeline for BERT). The performance of CMed-Word2Vec (Word-Level & Character-Level) and CMed-GloVe (Word-Level & Character-Level) is less than that of CMed-BERT and BERT. The average F1-Scores of CMed-Word2Vec are 68.01% and 64.33% for its Word-Level and Character-Level variants, respectively. The difference between the two is 3.68%, which is higher than the difference between CMed-BERT and BERT (i.e., 2.04%). These performances are poorer than those of CMed-BERT and BERT, but better than those of CMed-GloVe Word-Level (56.10%) and Character-Level (55.94%).

### *6.1.2. SAHSU*

Using the SAHSU dataset, the gap between the performance of similar language models in each downstream model is smaller than that using CCKS 2019. The average of F1-Score performance gap of models within the same class is always less than 1%. Precision, Recall, and F1-Score stats of the models, compared with those found using CCKS 2019, have shown better results (the overall average of F1-Score being 16% higher than with CCKS 2019). Compared with the general pre-training models, CMedLMs increased the performance of each downstream model by an average of 4.64%. Among them, the best-performing model is still CMed-BERT-ResGRU-CNN-CRF (improving upon the BERT-ResGRU-CNN-CRF pipeline by 2.21% and upon a fine-tuned BERT by 3.53% in their F1-scores), which demonstrates how CMed-BERT contributes to the performance on NER tasks.

Fig. 4 and Fig. 5 show how the F1-Scores of different pre-trained language models change during training and the difference in performance between these models. In summary, CMedLMs provide significant improvements in the perfor-

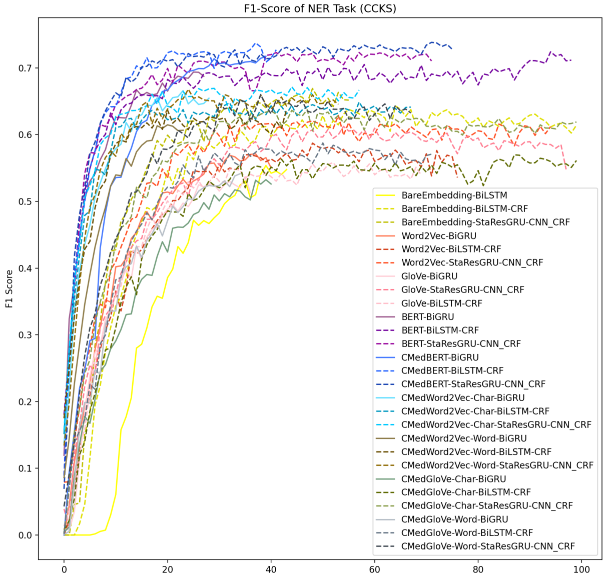

Figure 4: Performance of the top three pipelines under each type of embedding on the NER task (CCKS 2019)

mance of NER tasks. With the CCKS 2019 NER dataset, CMedLMs bring an additional 2.84% F1-Score improvement for each downstream model over general language models. On the SAHSU NER dataset, the performance showed a 4.64% increase. The best pipeline of CMed-BERT (which is CMed-BERT-ResGRU-CNN-CRF) shows improvements of 3.76% and 3.53% respectively on the two datasets compared with fine-tuned BERT models (Tab. 7).

### *6.2. Biomedical Text Classification*

The classification tests in this study use a dataset obtained and compiled from 4 mainstream Chinese biomedical Q&A and encyclopedia websites (see Tab. 2). This dataset can be split into 6 major categories. We tested a total of 45 pipelines based on 9 embeddings and a fine-tuned language model to evaluate the different language models' performance. These pipelines are composed of the following deep learning models: CNN-LSTM, CNN, BiLSTM, StaResGRU-CNN, DPCNN [65, 66, 67, 68]. The data analyzed include the

Table 7: Results of the Clinical Named Entity Recognition Task

| **Embedding** | **Model** | **CCKS 2019** | | | | | | **SAHSU** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Params | P | R | F1 | Time[1] | Loss[2] | Params | P | R | F1 | Time[1] | Loss[2] |
| Bare Embedding | BiLSTM_CRF | 3.17M | 0.6267 | 0.6601 | 0.6408 | 28 | 4.62/845 | 3.14M | 0.7804 | 0.8483 | 0.8129 | 28 | 2.87/874 |
| | CNN_LSTM | 1.34M | 0.4789 | 0.5876 | 0.5277 | 2 | 0.08/0.17 | 1.31M | 0.5907 | 0.7382 | 0.6562 | 2 | 0.06/0.18 |
| | BiLSTM | 2.66M | 0.5559 | 0.6195 | 0.5849 | 3 | 0.08/0.14 | 14.7M | 0.8181 | 0.8954 | 0.8543 | 3 | 0.06/0.13 |
| | BiGRU | 2.03M | 0.5557 | 0.6083 | 0.5801 | 3 | 0.08/0.15 | 2.00M | 0.7658 | 0.8622 | 0.8111 | 3 | 0.02/0.12 |
| | StaResGRU-CNN_CRF | 3.54M | 0.6547 | 0.6794 | 0.6645 | 37 | 17.97/741 | 2.51M | 0.8137 | 0.8770 | 0.8442 | 37 | 8.30/822 |
| BERT (Original) | Fine-Tuning | - | 0.7084 | 0.7540 | 0.7256 | - | - | - | 0.8375 | 0.8290 | 0.8511 | - | - |
| BERT (official) | BiLSTM_CRF | 15.2M | 0.6897 | 0.7716 | 0.7282 | 54 | 1.80/736 | 15.2M | 0.8297 | 0.8824 | 0.8549 | 54 | 2.01/955 |
| | CNN_LSTM | 17.8M | 0.6033 | 0.6910 | 0.6421 | 24 | 0.08/0.10 | 1.78M | 0.7518 | 0.8454 | 0.7953 | 15 | 0.03/0.07 |
| | BiLSTM | 14.7M | 0.6852 | 0.7640 | 0.7214 | 28 | 0.02/0.11 | 14.7M | 0.8181 | 0.8954 | 0.8543 | 19 | 0.01/0.08 |
| | BiGRU | 11.0M | 0.7047 | 0.7437 | 0.7221 | 27 | 0.02/0.09 | 11.0M | 0.8375 | 0.9037 | 0.8689 | 18 | 0.02/0.06 |
| | StaResGRU-CNN_CRF | 21.5M | 0.7214 | 0.7820 | 0.7499 | 63 | 0.99/930 | 21.5M | 0.8369 | 0.8944 | 0.8643 | 63 | 2.24/979 |
| Word2Vec (General) | BiLSTM_CRF | 3.45M | 0.5582 | 0.6493 | 0.5981 | 28 | 11.37/793 | 3.45M | 0.7912 | 0.8427 | 0.8159 | 28 | 2.09/969 |
| | CNN_LSTM | 1.22M | 0.4588 | 0.5502 | 0.4984 | 2 | 0.09/0.19 | 1.23M | 0.6145 | 0.7558 | 0.6772 | 2 | 0.04/0.16 |
| | BiLSTM | 2.93M | 0.4772 | 0.5290 | 0.5005 | 4 | 0.08/0.17 | 2.94M | 0.6742 | 0.7345 | 0.7021 | 3 | 0.06/0.12 |
| | BiGRU | 2.20M | 0.5699 | 0.5777 | 0.5725 | 3 | 0.06/0.14 | 2.20M | 0.7214 | 0.7872 | 0.7521 | 3 | 0.04/0.12 |
| | StaResGRU-CNN_CRF | 3.72M | 0.6217 | 0.6929 | 0.6551 | 38 | 1.39/913 | 3.72M | 0.8039 | 0.8659 | 0.8334 | 38 | 1.04/979 |
| GloVe (General) | BiLSTM_CRF | 3.04M | 0.5740 | 0.6069 | 0.5886 | 28 | 5.68/711 | 3.04M | 0.6906 | 0.7391 | 0.7116 | 28 | 5.02/811 |
| | CNN_LSTM | 1.20M | 0.4432 | 0.4903 | 0.4641 | 2 | 0.10/0.23 | 1.21M | 0.5598 | 0.6503 | 0.5984 | 2 | 0.07/0.14 |
| | BiLSTM | 2.52M | 0.4959 | 0.5038 | 0.4980 | 3 | 0.07/0.20 | 2.53M | 0.5939 | 0.6568 | 0.6210 | 3 | 0.07/0.13 |
| | BiGRU | 1.90M | 0.5400 | 0.5840 | 0.5600 | 3 | 0.07/0.18 | 1.90M | 0.6134 | 0.6688 | 0.6279 | 3 | 0.08/0.11 |
| | StaResGRU-CNN_CRF | 3.41M | 0.6055 | 0.6087 | 0.6056 | 37 | 6.97/906 | 3.41M | 0.7251 | 0.7586 | 0.7384 | 38 | 7.66/950 |
| CMed-BERT | BiLSTM_CRF | 15.2M | 0.7201 | 0.7874 | 0.7516 | 54 | 1.70/796 | 15.2M | 0.8331 | 0.8935 | 0.8618 | 54 | 0.78/1050 |
| | CNN_LSTM | 17.8M | 0.6481 | 0.7248 | 0.6839 | 18 | 0.06/0.11 | 1.78M | 0.7546 | 0.8472 | 0.7975 | 15 | 0.01/0.09 |
| | BiLSTM | 14.7M | 0.7095 | 0.7536 | 0.7303 | 22 | 0.02/0.11 | 14.7M | 0.8274 | 0.8991 | 0.8612 | 22 | 0.01/0.07 |
| | BiGRU | 11.0M | 0.7125 | 0.7635 | 0.7367 | 20 | 0.02/0.09 | 11.0M | 0.8407 | 0.9056 | 0.8716 | 18 | 0.02/0.08 |
| | StaResGRU-CNN_CRF | 21.5M | 0.7240 | 0.8072 | **0.7632** | 63 | 2.21/899 | 21.5M | 0.8669 | 0.9074 | **0.8864** | 63 | 1.06/951 |
| CMed-Word2Vec (Char-Level) | BiLSTM_CRF | 3.04M | 0.6646 | 0.7515 | 0.7043 | 28 | 1.97/930 | 3.04M | 0.7909 | 0.8575 | 0.8227 | 28 | 1.05/969 |
| | CNN_LSTM | 1.21M | 0.5390 | 0.6484 | 0.5869 | 2 | 0.09/0.16 | 1.22M | 0.6775 | 0.7956 | 0.7284 | 2 | 0.04/0.13 |
| | BiLSTM | 2.52M | 0.6435 | 0.7060 | 0.6731 | 3 | 0.03/0.15 | 2.53M | 0.7919 | 0.8566 | 0.8224 | 3 | 0.01/0.17 |
| | BiGRU | 1.90M | 0.6725 | 0.7609 | 0.7133 | 3 | 0.04/0.13 | 1.90M | 0.8186 | 0.8760 | 0.8459 | 3 | 0.01/0.14 |
| | StaResGRU-CNN_CRF | 3.41M | 0.6937 | 0.7555 | 0.7229 | 38 | 1.15/1036 | 3.41M | 0.8116 | 0.8816 | 0.8448 | 38 | 3.42/1100 |
| CMed-Word2Vec (Word-Level) | BiLSTM_CRF | 3.04M | 0.6549 | 0.7213 | 0.6859 | 28 | 1.04/843 | 3.04M | 0.7939 | 0.8585 | 0.8248 | 28 | 2.68/912 |
| | CNN_LSTM | 1.21M | 0.4877 | 0.5907 | 0.5340 | 2 | 0.07/0.16 | 1.21M | 0.6633 | 0.7863 | 0.7182 | 2 | 0.04/0.12 |
| | BiLSTM | 2.52M | 0.6045 | 0.6664 | 0.6330 | 3 | 0.04/0.14 | 2.53M | 0.7742 | 0.8464 | 0.8083 | 3 | 0.02/0.17 |
| | BiGRU | 1.90M | 0.6071 | 0.6686 | 0.6349 | 3 | 0.03/0.14 | 1.90M | 0.7773 | 0.8529 | 0.8125 | 3 | 0.02/0.12 |
| | StaResGRU-CNN_CRF | 3.41M | 0.6980 | 0.7636 | 0.7287 | 38 | 2.81/866 | 3.41M | 0.8535 | 0.8881 | 0.8701 | 37 | 2.02/871 |
| CMed-GloVe (Char-Level) | BiLSTM_CRF | 2.84M | 0.6223 | 0.6727 | 0.6462 | 28 | 0.88/941 | 2.84M | 0.7786 | 0.8353 | 0.8057 | 28 | 3.08/987 |
| | CNN_LSTM | 1.20M | 0.4193 | 0.5011 | 0.4554 | 2 | 0.11/0.17 | 1.20M | 0.6470 | 0.7660 | 0.7004 | 2 | 0.04/0.14 |
| | BiLSTM | 2.32M | 0.4857 | 0.5259 | 0.5046 | 3 | 0.08/0.17 | 2.32M | 0.6881 | 0.7660 | 0.7231 | 2 | 0.04/0.12 |
| | BiGRU | 1.74M | 0.5152 | 0.5727 | 0.5415 | 3 | 0.07/0.16 | 1.74M | 0.7180 | 0.8076 | 0.7595 | 3 | 0.04/0.10 |
| | StaResGRU-CNN_CRF | 3.26M | 0.6293 | 0.6907 | 0.6577 | 37 | 1.20/874 | 3.26M | 0.8098 | 0.8631 | 0.8355 | 38 | 1.48/895 |
| CMed-GloVe (Word-Level) | BiLSTM_CRF | 2.84M | 0.5965 | 0.6285 | 0.6113 | 28 | 2.08/730 | 2.84M | 0.7885 | 0.8353 | 0.8111 | 28 | 1.62/990 |
| | CNN_LSTM | 1.20M | 0.4151 | 0.4849 | 0.4465 | 2 | 0.10/0.18 | 1.20M | 0.6375 | 0.7613 | 0.6925 | 2 | 0.04/0.19 |
| | BiLSTM | 2.32M | 0.4772 | 0.5290 | 0.5005 | 3 | 0.09/0.18 | 2.32M | 0.6935 | 0.7539 | 0.7208 | 3 | 0.04/0.09 |
| | BiGRU | 1.74M | 0.5689 | 0.5849 | 0.5757 | 3 | 0.07/0.14 | 1.74M | 0.7287 | 0.7937 | 0.7573 | 3 | 0.04/0.08 |
| | StaResGRU-CNN_CRF | 3.26M | 0.6310 | 0.7001 | 0.6632 | 38 | 2.63/827 | 3.26M | 0.8166 | 0.8603 | 0.8373 | 38 | 4.92/995 |

[1] Time (s/epochs), [2] Loss/Valid Loss

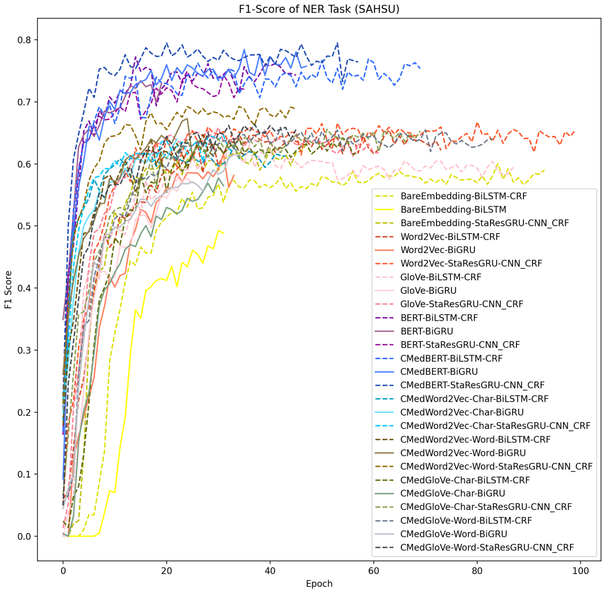

Figure 5: Performance of the top three pipelines under each type of embedding on the NER task (SAHSU)

number of trainable parameters, the time spent for each epoch, the F1-Score, and the final epoch's loss and validation set loss (more details in Tab. 8).

The experimental results show that the performance of the task-specific model is ideal in multi-class classification tasks, reaching an F1-Score of more than 90%. And the average of the F1-Scores of the models based on CMed-BERT reached 95.42%, which is 0.23% higher than the average of the F1-Scores of the BERT-based ones and also higher than fine-tuned BERT (95.28%). In addition, it can be seen that the effect of the models based on CMed-Word2Vec and CMed-GloVe on multi-class classification tasks is not much different than that of the models based on CMed-BERT and BERT. However, both the quantity of parameters and the epoch time is significantly lower than the BERT-based pipelines. Therefore, CMed-Word2Vec and CMed-GloVe have higher comprehensive competitiveness in classification tasks than CMed-BERT and BERT. At the same time, CMed-Word2Vec (char & word levels) and CMed-GloVe (char & word levels) show better performances than general Word2Vec and GloVe (with

Table 8: The Performance of Chinese Biomedical Text Classification

| Embedding | Model | Performance | | | | | | Embedding | Performance | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Params | P | R | F1 | Time[1] | Loss[2] | | Params | P | R | F1 | Time[1] | Loss[2] |
| GloVe (General) | CNN-LSTM | 0.06M | 0.9387 | 0.9383 | 0.9383 | 12 | 0.11/0.20 | Word2Vec (General) | 0.07M | 0.9464 | 0.9461 | 0.9460 | 12 | 0.11/0.18 |
| | CNN | 0.07M | 0.9333 | 0.9306 | 0.9304 | 8 | 0.04/0.25 | | 0.13M | 0.9392 | 0.9390 | 0.9389 | 8 | 0.02/0.26 |
| | BiLSTM | 2.52M | 0.9272 | 0.9268 | 0.9265 | 89 | 0.20/0.23 | | 2.93M | 0.9472 | 0.9470 | 0.9469 | 95 | 0.06/0.20 |
| | StaResGRU-CNN | 3.16M | 0.9550 | 0.9541 | 0.9542 | 110 | 0.11/0.16 | | 3.20M | 0.9537 | 0.9528 | 0.9528 | 110 | 0.08/0.17 |
| | DPCNN | 2.03M | 0.8288 | 0.6830 | 0.6450 | 97 | 0.02/0.81 | | 2.10M | 0.8706 | 0.7267 | 0.7413 | 100 | 0.02/0.86 |
| BERT (official) | CNN-LSTM | 0.35M | 0.9575 | 0.9573 | 0.9572 | 587 | 0.10/0.14 | CMed-BERT | 0.35M | 0.9603 | 0.9598 | 0.9599 | 573 | 0.10/0.14 |
| | CNN | 1.97M | 0.9513 | 0.9491 | 0.9494 | 512 | 0.05/0.16 | | 1.97M | 0.9520 | 0.9510 | 0.9509 | 660 | 0.04/0.30 |
| | BiLSTM | 14.7M | 0.9581 | 0.9579 | 0.9578 | 807 | 0.08/0.14 | | 14.7M | 0.9559 | 0.9555 | 0.9555 | 796 | 0.07/0.14 |
| | StaResGRU-CNN | 17.31M | 0.9615 | 0.9612 | 0.9612 | 1030 | 0.09/0.15 | | 17.30M | 0.9639 | 0.9616 | **0.9619** | 1118 | 0.09/0.15 |
| | DPCNN | 4.26M | 0.9408 | 0.9332 | 0.9339 | 1466 | 0.08/0.20 | | 4.26M | 0.9467 | 0.9431 | 0.9432 | 1522 | 0.09/0.24 |
| CMed-Word2Vec (Char-Level) | CNN-LSTM | 0.06M | 0.9398 | 0.9383 | 0.9386 | 13 | 0.06/0.24 | CMed-Word2Vec (Word-Level) | 0.06M | 0.9433 | 0.9431 | 0.9431 | 12 | 0.08/0.21 |
| | CNN | 0.07M | 0.9119 | 0.9107 | 0.9105 | 9 | 0.04/0.39 | | 0.07M | 0.9303 | 0.9299 | 0.9299 | 9 | 0.04/0.28 |
| | BiLSTM | 2.52M | 0.9431 | 0.9423 | 0.9424 | 80 | 0.02/0.27 | | 2.52M | 0.9391 | 0.9387 | 0.9385 | 81 | 0.01/0.29 |
| | StaResGRU-CNN | 3.16M | 0.9566 | 0.9561 | 0.9562 | 101 | 0.09/0.19 | | 3.16M | 0.9580 | 0.9580 | 0.9579 | 100 | 0.07/0.17 |
| | DPCNN | 2.03M | 0.9019 | 0.8814 | 0.8826 | 92 | 0.01/0.21 | | 2.03M | 0.9380 | 0.9339 | 0.9342 | 93 | 0.01/0.37 |
| CMed-GloVe (Char-Level) | CNN-LSTM | 0.05M | 0.9484 | 0.9477 | 0.9476 | 16 | 0.14/0.20 | CMed-GloVe (Word-Level) | 0.06M | 0.9425 | 0.9421 | 0.9421 | 18 | 0.12/0.19 |
| | CNN | 0.04M | 0.9321 | 0.9321 | 0.9320 | 16 | 0.04/0.26 | | 0.04M | 0.9297 | 0.9296 | 0.9296 | 16 | 0.05/0.23 |
| | BiLSTM | 2.31M | 0.9420 | 0.9420 | 0.9419 | 62 | 0.24/0.18 | | 2.31M | 0.9428 | 0.9422 | 0.9422 | 63 | 0.05/0.24 |
| | StaResGRU-CNN | 3.14M | 0.9545 | 0.9536 | 0.9538 | 82 | 0.14/0.17 | | 3.14M | 0.9542 | 0.9537 | 0.9537 | 82 | 0.13/0.16 |
| | DPCNN | 1.99M | 0.9279 | 0.9235 | 0.9232 | 64 | 0.06/0.44 | | 1.99M | 0.8889 | 0.8218 | 0.8248 | 60 | 0.02/0.51 |
| Bare Embedding | CNN-LSTM | 0.47M | 0.9229 | 0.9210 | 0.9211 | 13 | 0.08/0.29 | BERT (Original) Fine-Tuning | - | 0.9335 | 0.9728 | 0.9528 | - | - |
| | CNN | 0.48M | 0.9434 | 0.9430 | 0.9429 | 8 | 0.01/0.21 | | | | | | | |
| | BiLSTM | 2.93M | 0.9430 | 0.9425 | 0.9425 | 88 | 0.07/0.25 | | | | | | | |
| | StaResGRU-CNN | 3.58M | 0.9502 | 0.9496 | 0.9496 | 91 | 0.08/0.19 | | | | | | | |
| | DPCNN | 2.44M | 0.9127 | 0.9084 | 0.9070 | 64 | 0.01/0.27 | | | | | | | |

[1] Time (s/epochs), [2] Loss/Valid Loss

an increase of 2.08%, 3.55% and 6.08%, 3.96%, respectively). These results also prove that language models pre-trained based on domain knowledge can better solve NLP downstream tasks, including classification.

### 6.3. Free-form Text EMR-based Disease Diagnosis Prediction

In the experiments of the Free-form Text EMR-based Disease Diagnosis Prediction task, EMR data provided by the Second Affiliated Hospital of Soochow University and CCKS 2019 Task1 were both used for evaluation. We finally merged these two datasets into a new free-form text EMR-based Disease Diagnosis Prediction dataset named SAHSU-CCKS, which contains 2,808 records with 44 types of disease diagnosis labels (Tab. 9).

Experiments assess the performance of 86 pipelines that are respectively based on the permutation and combination of 1 language model fine-tuning, 1 Bare Embedding, 8 language models, and 10 mainstream deep learning models with the same parameters (64 for batch size, 100 epochs, the sequence length of 512, 10 for early stopping and 0.4 dropout rate, see Tab. 5 & 6). The results are shown in Tab. 9.

The performance of CMed-BERT on diagnosis prediction tasks is generally higher than that of BERT. However, there are two exceptions, namely BiGRU and CNN-LSTM, based on CMed-BERT and yet perform slightly worse than their BERT counterparts. CMed-BERT achieved an average of F1-Score of 62.77%, almost 4% higher than BERT's 58.84%. In the diagnosis prediction tasks, however, the best-performing models are not the same as for other tasks. In this latter case, the top 3 models are CMed-GloVe (Word-Level)-StaResGRU-CNN (76.60%), CMed-GloVe (Char-Level)-StaResGRU-CNN (74.50%) and CMed-Word2Vec (Char-Level)-StaResGRU-CNN (74.27%). For comparison, CMed-BERT-StaResGRU-CNN (73.48%) and BERT-StaResGRU-CNN (70.28%) only reached 73.48% and 70.28%, respectively. This shows how CMed-Word2Vec and CMed-GloVe are more performant in prediction tasks. From this densely distributed figure (Fig. 6), it is clear that these models showed a smaller F1-score gap during training in the prediction/classification task than in the NER task. Additionally, the CMedLMs model (with an average of F1-score of 57.2 6% per model) has a higher competitive performance on this mixed task than general language models (with an average of the F1-scores of 52.52% per model) and Bare Embedding (with an average of F1-score of 46.06% per model). This also reflects the importance of Chinese Biomedical domain knowledge in related Biomedical NLP tasks.

## 7. Analysis and Discussion

### *7.1. Analysis*

This study thoroughly evaluated the effect of pre-trained language models in various biomedical field tasks through sub-tasks such as Clinical NER, Biomedical Text Classification, Free-form Text EMR-based Disease Diagnosis Prediction. Upon analyzing the results, the best performing models in each

Table 9: Free-form Text EMR-based Disease Diagnosis Prediction Results of Comparison Models

| **Embedding** | **Model** | **Performance** | | | | | | **Embedding** | **Performance** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Params | P | R | F1 | Time[1] | Loss[2] | | Params | P | R | F1 | Time[1] | Loss[2] |
| GloVe (General) | CNN-LSTM | 0.06M | 0.2012 | 0.2412 | 0.2013 | 1 | 1.44/2.13 | Word2Vec (General) | 0.07M | 0.2247 | 0.2645 | 0.2315 | 1 | 1.26/2.07 |
| | CNN | 0.07M | 0.5540 | 0.5523 | 0.5293 | 1 | 0.22/1.43 | | 0.13M | 0.6319 | 0.6021 | 0.5950 | 1 | 0.07/1.03 |
| | BiGRU | 1.93M | 0.4818 | 0.4845 | 0.4648 | 5 | 0.10/1.93 | | 2.23M | 0.4260 | 0.4135 | 0.4002 | 5 | 0.08/2.17 |
| | CNN-GRU | 0.05M | 0.4419 | 0.4037 | 0.4040 | 1 | 0.45/1.95 | | 0.06M | 0.4941 | 0.4543 | 0.4407 | 1 | 0.59/1.75 |
| | BiLSTM | 2.56M | 0.3631 | 0.3592 | 0.3448 | 6 | 0.22/2.62 | | 2.96M | 0.3852 | 0.3908 | 0.3666 | 6 | 0.42/2.39 |
| | Dropout-BiGRU | 0.16M | 0.5619 | 0.5557 | 0.5309 | 3 | 0.87/1.54 | | 0.2M | 0.5996 | 0.6007 | 0.5792 | 3 | 0.75/1.33 |
| | Dropout-AVRNN | 0.17M | 0.4746 | 0.4963 | 0.4691 | 3 | 0.95/1.30 | | 0.20M | 0.6193 | 0.6101 | 0.5920 | 3 | 0.76/1.16 |
| | AVRNN | 0.26M | 0.6704 | 0.6803 | 0.6674 | 3 | 0.51/1.43 | | 0.30M | 0.5826 | 0.5966 | 0.5677 | 3 | 0.74/1.20 |
| | AVCNN | 0.82M | 0.6769 | 0.5995 | 0.6054 | 2 | 0.50/1.23 | | 1.12M | 0.6651 | 0.6361 | 0.6259 | 2 | 0.44/1.08 |
| | StaResGRU-CNN | 3.14M | 0.6316 | 0.6217 | 0.6113 | 7 | 0.80/1.20 | | 3.18M | 0.7174 | 0.6500 | 0.6457 | 7 | 0.82/1.19 |
| BERT (official) | CNN-LSTM | 0.35M | 0.4287 | 0.3762 | 0.3823 | 38 | 0.48/1.80 | CMed-BERT | 0.35M | 0.5731 | 0.5273 | 0.5285 | 39 | 0.30/1.69 |
| | CNN | 1.97M | 0.5544 | 0.5300 | 0.5103 | 37 | 0.09/1.26 | | 1.97M | 0.5921 | 0.5562 | 0.5448 | 37 | 0.06/2.26 |
| | BiGRU | 11.0M | 0.6091 | 0.5918 | 0.5741 | 49 | 0.03/1.31 | | 11.0M | 0.5971 | 0.5829 | 0.5631 | 47 | 0.04/1.38 |
| | CNN-GRU | 0.33M | 0.6296 | 0.5835 | 0.5791 | 43 | 0.13/1.21 | | 0.33M | 0.6073 | 0.5802 | 0.5745 | 41 | 0.14/1.68 |
| | BiLSTM | 14.7M | 0.6006 | 0.5696 | 0.5657 | 53 | 0.13/1.46 | | 14.7M | 0.6359 | 0.6259 | 0.6136 | 59 | 0.05/1.61 |
| | Dropout-BiGRU | 1.31M | 0.6561 | 0.6506 | 0.6310 | 39 | 0.32/1.56 | | 1.31M | 0.7063 | 0.6826 | 0.6730 | 41 | 0.20/1.35 |
| | Dropout-AVRNN | 1.17M | 0.6862 | 0.6585 | 0.6417 | 39 | 0.58/1.11 | | 1.33M | 0.6996 | 0.6861 | 0.6745 | 49 | 0.33/1.47 |
| | AVRNN | 1.33M | 0.6610 | 0.6701 | 0.6488 | 46 | 0.17/1.21 | | 1.33M | 0.6976 | 0.7158 | 0.6911 | 44 | 0.14/1.15 |
| | AVCNN | 9.74M | 0.6706 | 0.6605 | 0.6486 | 51 | 0.24/1.11 | | 9.74M | 0.6998 | 0.6971 | 0.6800 | 40 | 0.06/1.69 |
| | StaResGRU-CNN | 16.28M | 0.7389 | 0.7114 | 0.7028 | 68 | 0.23/1.24 | | 16.42M | 0.7577 | 0.7534 | 0.7348 | 68 | 0.28/1.20 |
| CMed-Word2Vec (Char-Level) | CNN-LSTM | 0.55M | 0.3164 | 0.3402 | 0.3159 | 1 | 0.54/1.77 | CMed-Word2Vec (Word-Level) | 0.06M | 0.3793 | 0.3989 | 0.3684 | 1 | 0.65/1.77 |
| | CNN | 0.07M | 0.5888 | 0.6029 | 0.5722 | 1 | 0.04/1.77 | | 0.07M | 0.6639 | 0.6298 | 0.6289 | 1 | 0.06/1.20 |
| | BiGRU | 1.93M | 0.4465 | 0.4673 | 0.4267 | 5 | 0.11/2.89 | | 8.14M | 0.4663 | 0.4835 | 0.4577 | 5 | 0.11/2.16 |
| | CNN-GRU | 2.56M | 0.4566 | 0.4420 | 0.4365 | 1 | 0.45/2.32 | | 0.05M | 0.5765 | 0.5212 | 0.5268 | 1 | 0.29/1.77 |
| | BiLSTM | 2.56M | 0.4685 | 0.4883 | 0.4568 | 5 | 0.15/2.07 | | 2.56M | 0.4186 | 0.4621 | 0.4236 | 5 | 0.15/2.17 |
| | Dropout-BiGRU | 10.16M | 0.6691 | 0.6649 | 0.6448 | 3 | 0.53/1.38 | | 0.16M | 0.6852 | 0.6789 | 0.6628 | 3 | 0.61/1.39 |
| | Dropout-AVRNN | 0.17M | 0.7132 | 0.6939 | 0.6903 | 3 | 0.78/1.01 | | 0.17M | 0.7629 | 0.7365 | 0.7278 | 3 | 0.60/1.02 |
| | AVRNN | 0.26M | 0.7457 | 0.7011 | 0.6989 | 4 | 0.38/1.42 | | 0.26M | 0.7187 | 0.7242 | 0.7040 | 4 | 0.16/1.52 |
| | AVCNN | 0.82M | 0.3787 | 0.5036 | 0.4077 | 2 | 1.46/1.72 | | 0.82M | 0.5989 | 0.6583 | 0.6147 | 2 | 1.11/1.32 |
| | StaResGRU-CNN | 11.14M | 0.7854 | 0.7473 | 0.7427 | 7 | 0.67/0.95 | | 11.14M | 0.7700 | 0.7280 | 0.7252 | 8 | 0.60/0.87 |
| CMed-GloVe (Char-Level) | CNN-LSTM | 0.06M | 0.2944 | 0.2923 | 0.2797 | 1 | 1.08/1.59 | CMed-GloVe (Word-Level) | 0.06M | 0.3564 | 0.3335 | 0.3197 | 1 | 1.01/1.62 |
| | CNN | 0.04M | 0.6278 | 0.6170 | 0.6052 | 1 | 0.14/1.08 | | 0.04M | 0.6423 | 0.6268 | 0.6174 | 1 | 0.05/1.62 |
| | BiGRU | 1.77M | 0.5138 | 0.5083 | 0.4883 | 5 | 0.04/1.88 | | 3.10M | 0.4520 | 0.4710 | 0.4432 | 5 | 0.04/3.03 |
| | CNN-GRU | 0.04M | 0.5240 | 0.5203 | 0.5017 | 1 | 0.29/1.26 | | 0.05M | 0.4512 | 0.4721 | 0.4369 | 1 | 0.22/1.55 |
| | BiLSTM | 2.35M | 0.4182 | 0.4007 | 0.3950 | 5 | 0.08/2.42 | | 2.35M | 0.3874 | 0.3845 | 0.3662 | 5 | 0.08/3.31 |
| | Dropout-BiGRU | 0.14M | 0.7091 | 0.7088 | 0.6896 | 3 | 0.37/1.35 | | 0.16M | 0.6852 | 0.6789 | 0.6628 | 3 | 0.36/1.27 |
| | Dropout-AVRNN | 0.15M | 0.5434 | 0.5496 | 0.5275 | 3 | 1.06/1.19 | | 0.15M | 0.6533 | 0.6396 | 0.6156 | 3 | 0.67/1.17 |
| | AVRNN | 0.25M | 0.6643 | 0.6639 | 0.6540 | 3 | 0.40/1.03 | | 0.25M | 0.6690 | 0.6491 | 0.6354 | 4 | 0.54/1.17 |
| | AVCNN | 0.67M | 0.7114 | 0.6942 | 0.6816 | 2 | 0.55/1.56 | | 0.67M | 0.7392 | 0.7014 | 0.6903 | 2 | 0.70/0.86 |
| | StaResGRU-CNN | 3.12M | 0.7809 | 0.7542 | 0.7450 | 7 | 0.40/1.08 | | 3.14M | 0.8019 | 0.7647 | **0.7660** | 7 | 0.88/1.29 |
| Bare Embedding | CNN-LSTM | 0.21M | 0.3260 | 0.3065 | 0.2931 | 1 | 1.129/1.89 | | | | | | | |
| | CNN | 0.22M | 0.5000 | 0.4979 | 0.4719 | 1 | 1.11/1.45 | | | | | | | |
| | BiGRU | 2.08M | 0.5100 | 0.5247 | 0.5000 | 5 | 0.20/2.37 | | | | | | | |
| | CNN-GRU | 0.20M | 0.1705 | 0.2049 | 0.1720 | 1 | 1.25/2.22 | | | | | | | |
| | BiLSTM | 2.70M | 0.3334 | 0.3436 | 0.3205 | 6 | 0.76/2.41 | BERT (Original) Fine-Tuning | | | | | | |
| | Dropout-BiGRU | 0.31M | 0.4934 | 0.4947 | 0.4812 | 3 | 0.78/1.54 | | - | 0.6429 | 0.6207 | 0.6316 | - | - |
| | Dropout-AVRNN | 0.32M | 0.6251 | 0.6123 | 0.6019 | 3 | 0.49/1.76 | | | | | | | |
| | AVRNN | 0.41M | 0.5405 | 0.5478 | 0.5347 | 3 | 0.52/2.13 | | | | | | | |
| | AVCNN | 0.97M | 0.6940 | 0.6497 | 0.6463 | 2 | 0.49/1.05 | | | | | | | |
| | StaResGRU-CNN | 3.29M | 0.5968 | 0.5924 | 0.5845 | 9 | 0.56/1.34 | | | | | | | |

[1] Time (s/epochs), [2] Loss/Valid Loss

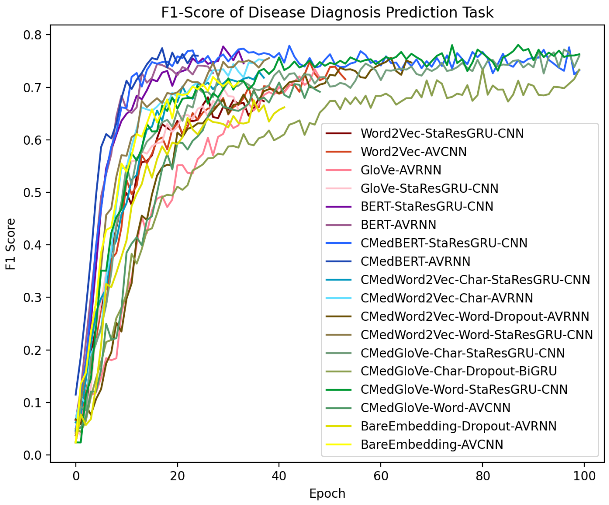

Figure 6: Performance of the top 2 models in each language model on the Free-form Text EMR-based Disease Diagnosis Prediction task

task are as follows: CMed-BERT-ResGRU-CNN-CRF (NER task, with CCKS 2019 dataset: F1 of 76.32%; with SAHSU dataset: F1 of 88.64%), CMed- BERT-StaResGRU-CNN (Classification task, F1: 96.19%), CMed-GloVe (Word-Level)-StaResGRU-CNN (Disease Diagnosis Prediction task, F1: 76.60%). In these parallel comparative experiments combining different language models, the effectiveness of the proposed StaResGRU-CNN model has been verified.

Comparing the average of the F1-scores of CMed-BERT with other bare embedding's pipelines shows that CMed-BERT's is consistently higher. Following are the performance gains in comparison with average F1-scores of bare embedding's pipelines: NER (CCKS: +13.35%, SAHSU: +5.99%), Classification (+2.16%) and Disease Diagnosis Prediction (+16.71%). The pipelines based on CMed-Word2Vec and CMed-GloVe have a similar performance with CMed-BERT and BERT in the Biomedical Classification and Disease Diagnosis Prediction tasks, but the number of parameters is smaller and each epoch takes a shorter time, thus providing higher efficiency. Finally, these models do not rely heavily on powerful hardware (e.g., GPU, TPU), making them more easily deployable than CMed-BERT and BERT. In addition, CMedLMs perform higher on average than general language models by 3.74% (2.84%, CCKS 2019 NER; 4.64%, SAHSU NER), 2.38% and 4.73% on the first three evaluation tasks, respectively. These gaps can also be observed as "stratification" in the training process visualization (Fig. 4-6). These performance improvements for multiple tasks help analyze large-scale datasets when deployed in online/streamed biomedical NLP tasks processing systems.

### *7.2. The role of StaResGRU-CNN and CMedLMs in predictive intelligence tasks*

The proposed method is used as a deep learning model for decision-making tasks with different granularities and can be used for various predictive tasks related to discrete variables (e.g., prediction/classification attribute labels). As an integrated application based on the above tasks, predictive medical intelligence can achieve greater practical goals (e.g., prediction and decision-making tasks in the medical field) by integrating these complex prediction and decision-making tasks. According to the granularity level of decision/prediction/classification in their tasks, these goals can be transformed into specific applications (e.g., token boundary decision and type classification in clinical Named Entity Recognition) according to the granularity level of decision/prediction/classification contained in their tasks. Therefore, in different predictive intelligence tasks, the proposed model can exert different actual effects in different scenarios or domains according to the characteristics of the inputted dataset.

In addition, simple learning of latent patterns in specific tasks from data-driven models can gradually evolve into learning a large amount of prior experience to realize multiple complex prediction tasks. This exciting change will greatly promote the effective use of valuable but not yet annotated data. At the same time, these language models that "absorb" the "wisdom" stored in the large-scale corpus can be directly customized or combined with downstream task models to achieve ideal performance in multi-type prediction and decision tasks better. Therefore, the pre-trained language models also play an important role in predictive intelligence tasks.

## 8. Conclusion

This study proposed a Stacked Residual Gated Recurrent Unit-Convolutional Neural Networks (StaResGRU-CNN) combined with the pre-trained SOTA language models for prediction tasks based on biomedical texts. Our proposed stacked model increases the depth of the network. This makes the model have better nonlinear expression capabilities, can learn more complex transformations, and can fit more diverse feature inputs. At the same time, it makes the model have more powerful expression ability and feature learning layer by layer. And it can also perform long-term time series prediction and avoid overfitting. And the residual layer solves the degradation problem of the deep neural network well and makes the model converge faster. All of these features contribute to the model achieving the ideal results in the above-mentioned biomedical text-related predictive tasks.

The work also explores some issues that have not yet been solved and presents practical difficulties in the field of Chinese biomedicine. It proposes and validates the first pre-trained language model series in the Chinese biomedical field in response to those unsolved issues. It also proposes a novel free-form text EMRs-based Disease Diagnosis Prediction to support intelligent clinical assistants' design. These models are made according to 3 schemes (BERT, Word2Vec, Glove). Through transfer learning, a language model can assimilate a large amount of biomedical knowledge in Chinese to generate word representations that are more suitable for the biomedical field and support downstream prediction tasks. And extensive comparative experiments presented have also proved their effectiveness.

In conclusion, it was proved that bringing biomedical domain knowledge into general language models improves their performance in biomedicine-related tasks. Together with the tested medical tasks, the proposed StaResGRU-CNN model and the presented set of language models provide a framework for building smarter and more accurate automated clinical assistants and moving towards more efficient and humane HCI-oriented medical services.

## 9. Limitation and Future Work

The models proposed in this study suffer from several resources and implementation issues, mainly: limited diversity of the medical data available, limitations in computing power (hardware limitations), models' inability to identify polysemy, and take advantage of abbreviations and implicit semantics. In the future, we will solve the problems mentioned above accordingly.

We will also incorporate the Autoregressive-based language model, including the GPT type (e.g., GPT-2, GPT-3), into a broader evaluation. A more detailed investigation will also be conducted on the progress of embedding-based and transformer-based language models in the fields of biomedicine and medical psychology. The relevant language models that have been pre-trained in biomedical and medical psychology domain knowledge will be applied to more complex or comprehensive tasks (e.g., Lifelong Machine Learning, dialogue system).

Besides, the variety and content of medical data are continuously updated as we expand our knowledge on emerging diseases (e.g., COVID-19). How to continuously and efficiently enable Chinese biomedical language models to learn comprehensively and deeply is the direction that future research needs to explore. Future explorative research needs to consider strategies to let Chinese biomedical language models learn in a continuous, efficient, comprehensive and deep fashion. This will also provide greater contributions to human-computer interaction-oriented medical predictive intelligence.

## Acknowledgment

This research is partly supported by VC Research (VCR 0000130) for Prof. Chang. At the same time, this study is also partially supported by the AI University Research Centre (AI-URC) through the XJTLU Key Program Special Fund (KSF-P-02, KSF-A-17). And this work has received support from the Suzhou Bureau of Science and Technology through the Key Industrial Technology Innovation Program (No. SYG201840). We also appreciate Google TensorFlow Research Cloud (TFRC) for providing support in computing resources.

In addition, we would like to thank all colleagues who participated in this research project, especially Ms. Yuming Li and Mr. Zhenjin Dai. We would also like to express our sincere thanks to Mr. Thomas Cilloni for providing English language support to the manuscript.

## References

[1] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234–1240.

[2] K. Huang, J. Altosaar, R. Ranganath, Clinicalbert: Modeling clinical notes and predicting hospital readmission, arXiv preprint arXiv:1904.05342.

[3] R. Liu, Y. Shi, C. Ji, M. Jia, A survey of sentiment analysis based on transfer learning, IEEE Access 7 (2019) 85401–85412.

[4] Y. Peng, S. Yan, Z. Lu, Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets, in: Proceedings of the 18th BioNLP Workshop and Shared Task, 2019, pp. 58–65.

[5] P. Ni, Y. Li, G. Li, V. Chang, Natural language understanding approaches based on joint task of intent detection and slot filling for iot voice interaction, Neural Computing and Applications (2020) 1–18.

[6] S. Ruder, Neural transfer learning for natural language processing, Ph.D. thesis, NUI Galway (2019).

[7] Y. Li, P. Ni, J. Peng, J. Zhu, Z. Dai, G. Li, X. Bai, A joint model of clinical domain classification and slot filling based on rcnn and bigru-crf, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 6133–6135.

[8] Q. Chen, X. Zhu, Z.-H. Ling, D. Inkpen, S. Wei, Neural natural language inference models enhanced with external knowledge, in: Proceedings of

the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 2406–2417.

[9] Q. Chen, X. Zhu, Z.-H. Ling, D. Inkpen, S. Wei, Natural language inference with external knowledge, arXiv preprint arXiv:1711.04289.

[10] R. Chatterjee, M. Negri, M. Turchi, M. Federico, L. Specia, F. Blain, Guiding neural machine translation decoding with external knowledge, in: Proceedings of the Second Conference on Machine Translation, 2017, pp. 157–168.

[11] C.-C. Huang, Z. Lu, Community challenges in biomedical text mining over 10 years: success, failure and the future, Briefings in bioinformatics 17 (1) (2015) 132–144.

[12] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, M. McDermott, Publicly available clinical bert embeddings, in: Proceedings of the 2nd Clinical Natural Language Processing Workshop, 2019, pp. 72–78.

[13] Q. Jin, B. Dhingra, W. W. Cohen, X. Lu, Probing biomedical embeddings from language models, NAACL HLT 2019 (2019) 82–89.

[14] A. Symeonidou, V. Sazonau, P. Groth, Transfer learning for biomedical named entity recognition with biobert, in: SEMANTICS Posters&Demos, 2019, pp. 1–5.

[15] C. Anagnostopoulos, K. Kolomvatsos, Predictive intelligence to the edge through approximate collaborative context reasoning, Applied Intelligence 48 (4) (2018) 966–991.

[16] J. Chapiro, J. S. Duncan, From code to bedside: Introducing predictive intelligence to interventional oncology, Radiology: Artificial Intelligence 1 (5) (2019) e190139.

[17] Y. Kathidjiotis, K. Kolomvatsos, C. Anagnostopoulos, Predictive intelligence of reliable analytics in distributed computing environments, Applied Intelligence 50 (2020) 3219–3238.

[18] Y. Li, P. Ni, V. Chang, Application of deep reinforcement learning in stock trading strategies and stock forecasting, Computing (2019) 1–18.

[19] P. Ni, Y. Li, G. Li, V. Chang, A hybrid siamese neural network for natural language inference in cyber-physical systems, ACM Transactions on Internet Technology (TOIT) 21 (2) (2021) 1–25.

[20] M. Gridach, A framework based on (probabilistic) soft logic and neural network for nlp, Applied Soft Computing 93 (2020) 106232.

[21] F. Gargiulo, S. Silvestri, M. Ciampi, G. De Pietro, Deep neural network for hierarchical extreme multi-label text classification, Applied Soft Computing 79 (2019) 125–138.

[22] D. Wang, P. Tiwari, S. Garg, H. Zhu, P. Bruza, Structural block driven enhanced convolutional neural representation for relation extraction, Applied Soft Computing 86 (2020) 105913.

[23] Y. Li, P. Ni, G. Li, V. Chang, Effective piecewise cnn with attention mechanism for distant supervision on relation extraction task., in: COMPLEXIS, 2020, pp. 53–60.

[24] P. Ni, Y. Li, V. Chang, Research on text classification based on automatically extracted keywords, International Journal of Enterprise Information Systems (IJEIS) 16 (4) (2020) 1–16.

[25] S. Tomori, T. Ninomiya, S. Mori, Domain specific named entity recognition referring to the real world by deep neural networks, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 236–242.

[26] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of NAACL-HLT, 2016, pp. 260–270.

[27] A. Bharadwaj, D. Mortensen, C. Dyer, J. Carbonell, Phonologically aware neural model for named entity recognition in low resource transfer settings, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1462–1472.

[28] G. R. Venkataraman, A. L. Pineda, O. J. Bear Don't Walk IV, A. M. Zehnder, S. Ayyar, R. L. Page, C. D. Bustamante, M. A. Rivas, Fastag: Automatic text classification of unstructured medical narratives, PLoS one 15 (6) (2020) e0234647.

[29] J. R. Quinlan, Simplifying decision trees, International journal of man-machine studies 27 (3) (1987) 221–234.

[30] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[31] M. M. Mirończuk, J. Protasiewicz, A recent overview of the state-of-the-art elements of text classification, Expert Systems with Applications 106 (2018) 36–54.

[32] M. Kamkarhaghighi, M. Makrehchi, Content tree word embedding for document representation, Expert Systems with Applications 90 (2017) 241–249.

[33] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, C. A. Iglesias, Enhancing deep learning sentiment analysis with ensemble techniques in social applications, Expert Systems with Applications 77 (2017) 236–246.

[34] J. Li, J. Li, X. Fu, M. A. Masud, J. Z. Huang, Learning distributed word representation with multi-contextual mixed embedding, Knowledge-Based Systems 106 (2016) 220–230.

[35] S. Sun, C. Luo, J. Chen, A review of natural language processing techniques for opinion mining systems, Information fusion 36 (2017) 10–25.

[36] J. Du, C.-M. Vong, C. P. Chen, Novel efficient rnn and lstm-like architectures: Recurrent and gated broad learning systems and their applications for text classification, IEEE transactions on cybernetics 51 (3) (2020) 1586–1597.

[37] J. Kim, S. Jang, E. Park, S. Choi, Text classification using capsules, Neurocomputing 376 (2020) 214–221.

[38] A. M. Dai, Q. V. Le, Semi-supervised sequence learning, in: Advances in neural information processing systems, 2015, pp. 3079–3087.

[39] T. Miyato, A. M. Dai, I. Goodfellow, Adversarial training methods for semi-supervised text classification, arXiv preprint arXiv:1605.07725.

[40] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 655–665.

[41] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of NAACL-HLT, 2018, pp. 2227–2237.

[42] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 328–339.

[43] P. Ni, Y. Li, J. Zhu, J. Peng, Z. Dai, G. Li, X. Bai, Disease diagnosis prediction of emr based on bigru-att-capsnetwork model, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 6166–6168.

[44] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[45] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[46] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.

[47] I. Tenney, D. Das, E. Pavlick, Bert rediscovers the classical nlp pipeline, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4593–4601.

[48] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2267–2273.

[49] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[50] M. Peters, W. Ammar, C. Bhagavatula, R. Power, Semi-supervised sequence tagging with bidirectional language models, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1756–1765.

[51] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, H. Wang, Ernie 2.0: A continual pre-training framework for language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 8968–8975.

[52] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding,

in: Advances in neural information processing systems, 2019, pp. 5754–5764.

[53] Y. Li, B. Yu, X. Mengge, T. Liu, Enhancing pre-trained chinese character representation with word-aligned attention, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3442–3448.

[54] V. Yadav, S. Bethard, A survey on recent advances in named entity recognition from deep learning models, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2145–2158.

[55] A. Rios, R. Kavuluru, Convolutional neural networks for biomedical text classification: application in indexing biomedical articles, in: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, 2015, pp. 258–267.

[56] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991.

[57] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of machine learning research 12 (Aug) (2011) 2493–2537.

[58] M. Gridach, Character-level neural network for biomedical named entity recognition, Journal of biomedical informatics 70 (2017) 85–91.

[59] J. P. Chiu, E. Nichols, Named entity recognition with bidirectional lstm-cnns, Transactions of the Association for Computational Linguistics 4 (2016) 357–370.

[60] K. Hakala, S. Pyysalo, Biomedical named entity recognition with multilingual bert, in: Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, 2019, pp. 56–61.

[61] D. Dligach, M. Afshar, T. Miller, Toward a clinical text encoder: pretraining for clinical natural language processing with applications to substance misuse, Journal of the American Medical Informatics Association 26 (11) (2019) 1272–1278.

[62] A. Holzinger, J. Schantl, M. Schroettner, C. Seifert, K. Verspoor, Biomedical text mining: state-of-the-art, open problems and future challenges, in: Interactive knowledge discovery and data mining in biomedical informatics, Springer, 2014, pp. 271–300.

[63] Wikipedia, Offline Wikipedia (Chinese) download page, last accessed on February 21, 2020 (2020).
URL `https://dumps.wikimedia.org/zhwiki/latest/`

[64] Y. Song, S. Shi, J. Li, H. Zhang, Directional skip-gram: Explicitly distinguishing left and right context for word embeddings, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 175–180.

[65] J. Wang, L.-C. Yu, K. R. Lai, X. Zhang, Dimensional sentiment analysis using a regional cnn-lstm model, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 225–230.

[66] P. Liu, X. Qiu, X. Huang, Adversarial multi-task learning for text classification, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1–10. `doi:10.18653/v1/P17-1001`.
URL `https://www.aclweb.org/anthology/P17-1001`

[67] R. Johnson, T. Zhang, Deep pyramid convolutional neural networks for text categorization, in: Proceedings of the 55th Annual Meeting of the

Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 562–570.

[68] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, I. Androutsopoulos, Large-scale multi-label text classification on eu legislation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 6314–6322.