Neural Processing Letters (2023) 55:1209–1229
https://doi.org/10.1007/s11063-022-10933-3

A Multi-Task BERT-BiLSTM-AM-CRF Strategy for Chinese Named Entity Recognition

Xiaoyong Tang [1] · Yong Huang [1] · Meng Xia [1] · Chengfeng Long [2]

Accepted: 13 June 2022 / Published online: 12 July 2022

Abstract

Named entity recognition aims to identify and mark entities with specific meanings in text. It is a key technology to further extract entity relationships and mine other potential information in natural language processing. At present, the methods based on machine learning and deep learning have been widely used in the research of named entity recognition, but most learning models use feature extraction based on word and character level. The word preprocessing of this kind of model often ignores the context semantic information of the target word and can not realize polysemy. In addition, the loss of semantic information and limited training data greatly limit the improvement of model performance and generalization ability. In order to solve the above problems and improve the efficiency of named entity recognition technology in Chinese text, this paper constructs a multi-task BERT-BiLSTM-AM-CRF intelligent processing model, uses Bert to extract the dynamic word vector combined with context information, and inputs the results into CRF layer for decoding after further training through BiLSTM module. After attention mechanism network, the model can learn together on two Chinese datasets, Finally, CRF classifies and extracts the observation annotation sequence to get the final result. Compared with many previous single task models, the F1 score of this multi-task model in MASR and people's daily datasets has been significantly improved (0.55% and 3.41%), which demonstrates the effectiveness of multi-task learning for Chinese named entity recognition.

Keywords Named entity recognition · Multi-task learning · BERT · BiLSTM · AM

✉ Meng Xia
summerm_999@163.com

Xiaoyong Tang
tangxy@csust.edu.cn

Yong Huang
843923647@qq.com

Chengfeng Long
elong@hunau.edu.cn

[1] School of Computer and Communications Engineering, Changsha University of Science & Technology, Changsha 410114, Hunan, China

[2] School of Information and Intelligence, Hunan Agricultural University, Changsha 410128, Hunan, China

1 Introduction

Text mining is a challenging research topic at present [1, 2]. It can obtain valuable information and knowledge from massive text data. One of the famous technologies is Named Entity Recognition (NER), which aims to automatically identify specific entities in the input text, as well as the output location information and entity type of related entities [3–5]. It is an indispensable basic task in text mining, which affects the execution of a series of downstream tasks, such as entity-relationship extraction and text matching.

In recent years, many deep learning models, such as RNN, have been widely used in the field of text mining [6, 7]. Deep learning with word embedding can greatly improve the recognition ability of named entities [4, 8]. In this method, tokens are first mapped from a discrete one-hot representation to a low-dimensional space and become a dense embedding. Then, the embedding sequence of the sentence is input into the RNN, which uses a neural network to automatically extract features. Finally, the Softmax predicts the label of each token and makes the training of the model as an end-to-end process. Compared with the CRF model, the RNN model is very powerful in sequence modeling and can capture long-distance context information, which has the ability of neural networks to fit nonlinearities. However, the RNN has the problem of gradient disappearance and gradient explosion during long-distance training. Therefore, the long short-term memory model (LSTM) is proposed to deal with the longer sequences. A simple adjustment technique for the LSTM unit in RNN can significantly reduce overfitting [9]. After further improvement, a neural network model combining Bidirectional Long Short-Term Memory (BiLSTM) and Conditional Random Field (CRF) is proposed in paper [10], which is used for NER and part-of-speech tagging. The effect is better than simple CRF and BiLSTM. This two-way structure can obtain contextual sequence information, and it is widely used in tasks such as named entity recognition.

However, the disadvantage of these methods including RNN, LSTM, BiLSTM is that the labeling process for each token is carried out independently. And, the above predicted label cannot be used directly that may lead to the predicted label sequence may be invalid. On the other hand, CRF calculates a joint probability, optimizing the entire sequence (the final goal), rather than concatenating the best at each moment. At this point, CRF is better than these methods.

In order to do so, many modified CRF models are proposed for sequence labeling. The output layer of the neural network is connected to the CRF layer (the focus is to use the label transition probability) to do sentence-level label prediction. Therefore, the labeling process is no longer an independent classification of each token. In view of this, the BiLSTM-CRF model is proposed and can achieve remarkable results in sequence labeling tasks [11, 12].

Most of the deep models including BiLSTM-CRF use Word2Vec to perform feature extraction to obtain word embedding [12, 13]. There is a problem that these methods can not represent polysemy, because they mainly focus on the feature extraction between words, characters or words, and ignore the context or semantics of word context. Therefore, only static word vectors without context information are extracted, which leads to the decline of its entity recognition ability. In order to solve this problem, the Google team Jacob Devlin et al. proposed a BERT (Bidirectional Encoder Representation from Transformers) language preprocessing model to obtain word embedding [14]. As an advanced pre-training word vector model, BERT further enhances the generalization ability of the word vector model, fully describing character-level, word-level, sentence-level and even inter-sentence relationship features, and better characterizes the syntax and semantics in different contexts information.

In addition to deep learning, some research work are now also beginning to use transfer learning and multi-task learning for named entity recognition, but the recognition effect of these methods need to be further improved. In recognition of this, this work focus on the incorporating the self-attention mechanism and multi-task learning into the BERT-BiLSTM-CRF method to improve the accuracy of named entity recognition. We present threefold fundamental contributions, which can be summarized as follows:

- First, we construct a single-task model based on BERT-BiLSTM-CRF for named entity recognition. The dynamic word vector combined with context information is extracted by BERT. After further training by BiLSTM module, the result is input into CRF layer for decoding. CRF classifies and extracts the obtained observation annotation sequence to obtain the final result.
- Second, we incorporate the multi-task learning and self-attention mechanism into the BERT-BiLSTM-CRF model. After extracting the word vector through BERT and inputting it into BiLSTM for feature learning, through the attention mechanism network, the model can learn together on two Chinese datasets, and finally get the training results through CRF layer constraints.
- Finally, we evaluate the performance of our proposed multi-task BERT-BiLSTM-AM-CRF. The experimental results demonstrate that our proposed model can improve named entity recognition accuracy on the People's Daily dataset.

The rest of the paper is organized as following: Sect. 2 reviews the related studies in this domain. In Section 3, we describe some basic concepts of BERT, BiLSTM, and CRF. Section 4 proposes a single-task model base on BERT-BiLSTM-CRF. In Section 5, we present the multi-task model combining the self-attention mechanism and BERT-BiLSTM-CRF. We evaluate our proposed model and analyze the experiment results in Section 6. Finally, we conclude the paper in Section 7.

2 Related Work

Text mining is a process of extracting useful information from text so that we can focus on extracting specific text. Many intelligent information extraction systems are propsed to help identify important entities in text based on dictionary and rule methods [15, 16]. For example, the identification of protein and gene names [17, 18]. The researchers built models by integrating language models and statistical machine learning algorithms, such as hidden Markov model (HMM), support vector machine (SVM), and so on [19]. These models have shown good results in the GENIA corpus. However, HMM can only be limited to some contextual features. Some researchers proposed the Conditional Random Field Model (CRF) to address this issue well [20, 21]. CRF does not have the strict independence assumptions as HMM and it can accommodate arbitrary context information and the design is more flexible. However, CRF still has shortcomings in capturing long-distance dependent information.

In the past few year, many text mining methods are successfully applied in entity naming recognition for English text [5]. Some of these deep learning models are also extended to Chinese named entity recognition [10, 11, 22–24]. For example, Liu X.et al. applied convolutional neural network (CNN) to the named entity recognition of Chinese online medical inquiry texts [22]. In paper [11], a lexical feature based BiLSTM-CRF was constructed to conduct named entity recognition tasks in free-text section of Chinese adverse drug event reports. Ma X. and Hovy E.K. proposed an end-to-end method combining B-LSTMs, CNN, and CRF to solve the NER task [10]. De Oliveira D.M.et al. proposed a filter-stream named entity recognition, which uses 5 different filters to complete the calculation of the probability of named entities [23]. Jia Y.et al. incorporated convolutional neural network (CNN) into BiLSTM-CRF to build CNN-BiLSTM-CRF model, which can reach up to 91.03% F1 score on MSRA [25]. These work have a certain effect on Chinese named entity recognition.

However, because the entity boundary of Chinese named entity recognition is difficult to divide, Chinese NER still faces many challenges, such as Chinese word segmentation errors, out-of-vocabulary (OOV), *etc*. These challenges are often difficult to define the composition of words. Zhao S. et al. applied a LATTICE-LSTM-CRF method in Chinese clinical named entity recognition and obtain good performance [26]. Chang N. et al. incorporated the self-attention mechanism into the Chinese named entity recognition model to compute the weighted sum of character and word vectors. This method integrated the semantic features of the two representations (Mixed semantic model) that can achieve an $F1$ value of 95.36% on the MSRA dataset [27]. In paper [28], the authors compare several neural models (including BiLSTM) along with two pre-trained language models (word2vec and BERT) based on the Chinese EHR offered by CCKS 2019. The results show that BERT-BiLSTM-CRF model can achieve approximately 75% F1 score, which outperforms all other models during the tests.

To further improve the performance of the NER model, researchers continue to put forward various attempts, such as integrating various latest methods into existing tools to improve model performance [29]. However, how to further improve the performance of the Chinese text NER system based on the existing methods, especially in the case of limited datasets, and how to further improve the performance of NER is still a worthy research field.

One of the research directions is multi-task learning (MTL), which improves the performance of the model on a single dataset by using multiple related labeled datasets to train the target model [30]. Because the relevant dataset may contain effective supplementary information, it can help the model to more effectively solve the task on a single dataset after joint training.

The information contained in the relevant field datasets often has strong relevance and interaction. It can be guessed that the results of the model learning on one dataset may help adjust the model to perform better on other datasets. For example, a model trained by a researcher using the target dataset and three other related datasets successfully identified the unlabeled entities in the target dataset [31]. If a model can use relevant information outside of the training data, it is conceivable that it may perform better due to access to this additional information. Therefore, a model trained with multiple datasets may be better than a model trained separately achieve better results.

3 Method

3.1 BERT Pre-training Module

The internal structure of Bert is mainly composed of two-layer transformer stacked by encoder. The encoder structure includes multi head attention mechanism layer, feedforward neural network and merging normalization layer. Bert describes the scale through the following parameters: L, H, $A$and T, where L is the number of layers of the transformer,H represents the dimensionality of the output, A denotes the number of multi-head attention, and $T$is the total amount of all parameters of the model. The BERT model uses two unsupervised prediction tasks: Masked Language Model (Masked LM) and Next Sentence Prediction (NSP). In general, BERT Base is used as a baseline model in research work. The BERT Large

Table 1 Specific parameters of two BERT model

	L	H	A	T
$BERT\ Base$	12	768	12	110M
$BERT\ Large$	24	1024	16	340M

is derived from BERT Base and far exceeds it in term of performance. However, the hardware resource consumption of BERT Large is huge. The comparison of them is shown in Table 1. The difference between the two versions of the BERT model is only the parameter settings. Therefore, we use the BERT Base in the following performance evaluation.

As a milestone progress, Word2vec has had a huge impact on the development of NLP [13]. However, Word2vec itself is a shallow structure and the semantic information learned by its trained word vectors is restricted by the window size. The masked is similar to cbow of Word2vec, but unlike CBOW's full detection of words, masked randomly masks 15% of the tags in each sequence during the training process, and its goal is to predict the original vocabulary of the masked word based on the context. Unlike the pre-training of the language model from left to right, the representation learned by Masked LM can fuse the context of the left and right sides. The two-way Transformer in the model does not know which words it will be asked to predict, or which words have been replaced by random words, so it must maintain a distributed contextual representation for each input word. In addition, random replacement only occurs in 1.5% of all words, which will not affect the model's understanding of the language. Skip-gram similar to Word2vec predicts the context through the central word. NSP means that when the language model is pre-trained, two sentences are selected in two situations, one is to select two sentences in the corpus that are connected in real order; the other is One is that the second sentence is randomly selected from the corpus and spelled after the first sentence. In addition to the masked language model task mentioned above, let the model do a sentence relationship prediction, and use the Transformer model to determine whether the second sentence is a follow-up sentence of the first sentence.

We uses BERT instead of Word2vec to obtain text abstract features and uses BiLSTM to extract the contextual features corresponding to the predicted labels of the output text. Finally, the AM and CRF layer is constrained to output the optimal label sequence.

3.2 BiLSTM Module

To solve the problem of gradient disappearance and gradient explosion in long-sequence training tasks, the LSTM model proposed in paper introduced a memory unit and a threshold mechanism to realize the effective use of long-distance information. The unit and threshold mechanism has been improved to improve efficiency. In work [32], the authors use the improved threshold mechanism to effectively use the feature information before and after the input. The classical structure diagram of LSTM is shown in Fig. 1.

Where the σ is the activation function, W is the weight matrix, b is the bias vector, z_t is the content to be added, c_t is the update state at time t, i_t, f_t, and o_t are the output of the input gate, forget gate, and output gate, respectively. As result, h_t is the output of the entire LSTM unit at time t. The calculation formula of LSTM is as follows,

$$\begin{aligned} i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \\ z_t &= tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \end{aligned}$$

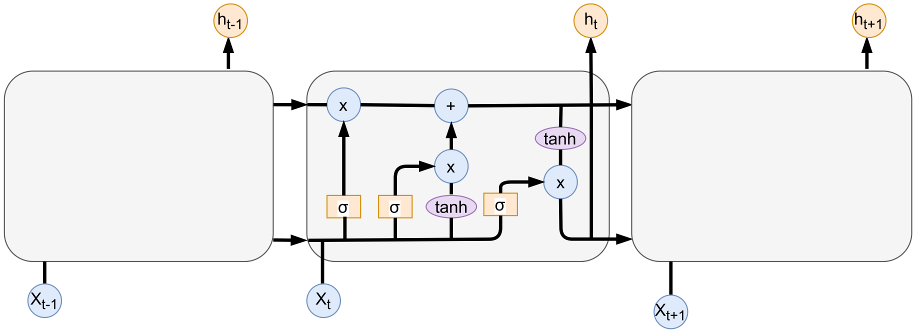


Fig. 1 The structure diagram of LSTM

$$\begin{aligned} c_t &= f_t c_t + i_t z_t \\ o_t &= tanh(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\ h_t &= o_t\ tanh(c_t) \end{aligned} \tag{1}$$

3.3 CRF Module

The output of BiLSTM will be used as the input of the CRF layer, and the final prediction result will be obtained by learning the order dependency information between tags. In the BiLSTM-CRF model, BiLSTM is good at processing long-distance text information, but it cannot handle the dependencies between adjacent labels. If the maximum value of the tag probability output by the BiLSTM is directly taken as the final prediction output, situations such as I as the beginning word and two consecutive B words may occur, so the model effect will be reduced. The CRF layer can modify the output of the BiLSTM layer by learning the transition probability between tags in the dataset, to ensure the rationality of the predicted tags and effectively avoid the occurrence of similar situations. Therefore, compared with a single BiLSTM model, by increasing the CRF The dependence information between the layer learning tags can significantly improve the model effect.

4 Single-Task Model

In this model, the text is first input into BERT for pre-training and the word vector with context information is extracted. Then, the double-layer LSTM model is input for sequence annotation. CRF classifies and extracts the obtained observation annotation sequence to obtain the final result. Figure 2 shows the overall flow diagram of the single-task model base on BERT-BiLSTM-CRF.

For any sequence, the segmentation text sequence is obtained by word segmentation. We can mask part of the words in the segmentation sequence. Here, the beginning of the sequence is marked with a special marker [CLS], and the sentences are separated with a marker [SEP]. In this way, the embedding composed of token embedding, segment embedding, and position embedding is obtained. At this time, the output embedding of each word in the sequence vector is input to the bidirectional transformer for feature extraction. Therefore, we can obtain the sequence vector containing context semantic features.

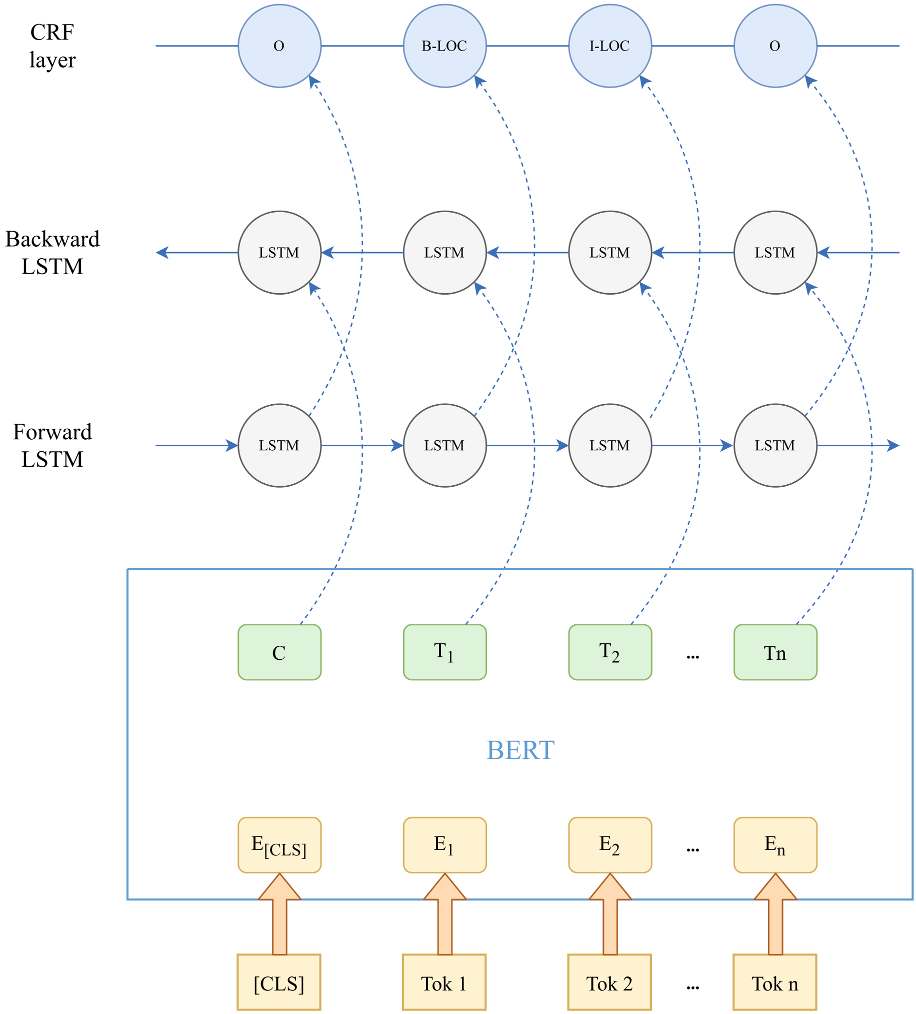


Fig. 2 Single-task model base on BERT-BiLSTM-CRF

The key part is the transformer structure, which abandons the CNN and RNN used in the previous deep learning tasks. The encoder-decoder architecture is also used in the transformer model like the attention model. But its structure is more complex than attention. In the transformer, the encoder layer is stacked by two encoders and the decoder layer is stacked by three encoders, which is shown in Fig. 3. In the BERT, the encoder layer is stacked by six encoders and so is the decoder layer. In a word, the text is input into BERT for pre-training that the word vector with context information is extracted. And then the double-layer LSTM model is input for sequence annotation, and the results are obtained through the CRF layer.

The encoder consists of two layers, a self-attention layer, and a feed forward neural network. Self-attention can help the current node not only focus on the current words but also get the semantics of the context. The coder includes not only the two layers network mentioned above, but also an attention layer in the middle of the two layers to help the current node obtain the current focus.

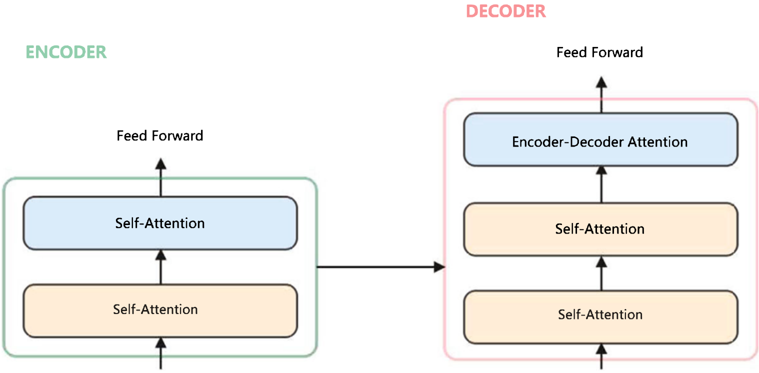


Fig. 3 The internal simplified structure of encoder and decoder

First, the model needs to perform an embedding operation on the input data (which can also be understood as an operation similar to W2C). Then, it is input to the encoder layer. After self-attention processes the data, it sends the data to the feed forward neural network. The calculation of the feed forward neural network can be parallel, and the output will be input to the next encoder. The internal details and data manipulation procedures for encoder are shown in Fig. 4.

In this model, the self-attention accountant calculates three new vectors, where the dimension of the vector is set as 512. Here, we call these three vectors as Query, Key, and Value. These vectors are the result of multiplying an embedding vector by a matrix. The matrix is randomly initialized, and the dimension is (64, 512). Note that the second dimension needs to be the same as the embedding dimension. The values are always updated in the process of BP, and the dimensions of these three vectors are lower than those of embedding.

Calculate the score of self-attention, which determines how much attention we pay to the rest of the input sentence when we encode a word in a certain position. This fractional value is calculated by multiplying Q and K .Next, divide the result of the point by a constant. Then we do a $Softmax$ calculation of the result. The result is the relevance of each word to the word in the current position. Generally, the square root of the first dimension d_k of the matrix mentioned above is adopted.

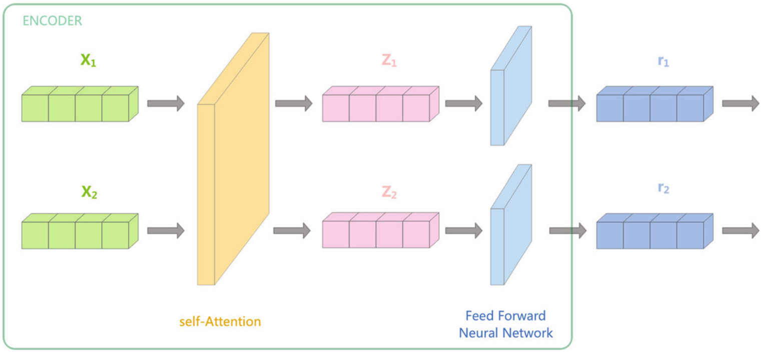


Fig. 4 The internal details of the model

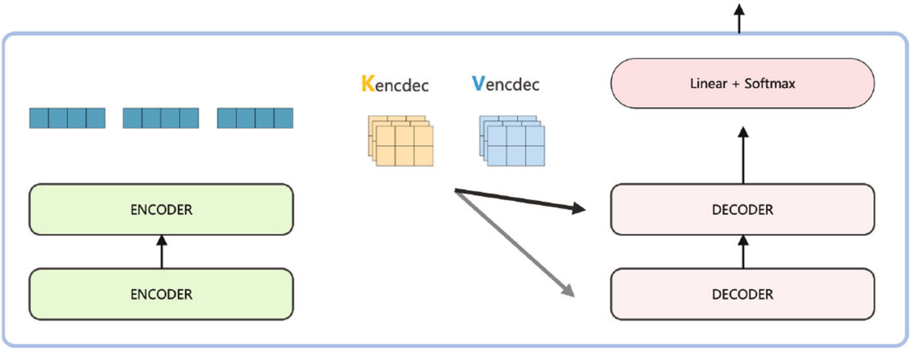


Fig. 5 The attention vectors K and V in model

The mechanism of multi-Attention is to project Q, K, and V through several different linear transformations, and finally, put the different attention results together. So the model can get the location information in different spaces, where W is the weight matrix formula as follows Eqs. (2) and (3).

For decoder, the part is similar to the encoder, but a masked multi head attachment is added at the bottom. As shown in the Fig. 5, the encoders layer output into a set of attention vectors K and V. Each decoder will use these attention vectors in its "encoder-decoder attention" layer, which helps the decoder focus on the appropriate position in the input sequence.

Finally, the vector can be mapped to the required words through a fully Connected layer and Softmax layer, when the decoder layer is finished. If our dictionary is n words, Softmax will input the probability of n words, and the corresponding word with the highest probability value is the final result.

The extracted word vector is input into the BiLSTM model, the context feature is extracted for sequence annotation, and the prediction label corresponding to the text is output. The output of BiLSTM is used as the input of the CRF layer, and the final prediction result is obtained by learning the sequence dependence information between tags.

The biggest advantage of this model is that BERT can combine the semantic information of the context for pre-training, and the semantic information characteristics of the semantic information that can be learned to the word level, the characteristics of the syntactic structure and the context through Transformer, and the model performance is improved by comparing with the pre-training model of the extraction of word vector features by CNN or RNN. At the same time, BiLSTM is used to further process the word vector, combined with the advantages of CRF, to further improve the effect of Chinese entity recognition. Then, we fine-tune the model parameters until the results of the model cannot continue to be optimized. While it is often possible to achieve acceptable performance for the model in this way, because our focus is on a single task, we ignore other information that may help optimize metrics. Specifically, this information comes from the training signals of the task in question. By sharing characterizations between related tasks, we can make our model better generalize the original tasks. In the "Multi-task Model" section below the article, we will further optimize the model in conjunction with multi-task learning.

5 Multi-Task Model

This model makes use of the fact that the results of learning on one dataset may help the model perform better on other datasets, which is described in the "Background" section. The model combines the two single-task models described in the "single-task model" section.

Multi-task advantage of the fact that learning on one dataset can help the model perform better on other datasets, as described in the Background section. Rich Caruana, on the other thing, points out that the goal of multi-task learning is to improve generalization capabilities by leveraging information that is included in specific areas of the relevant task training signal.

In deep learning, multi-task is usually done by hiding the Hard parameter sharing of the layer. Sharing Hard parameters reduces the risk of overfitting, typically by sharing hidden layers across all tasks while retaining the output layers for several specific tasks. The hard parameter sharing structure mode is shown in Fig. 6.

The joint study of NER tasks of different entities mainly embodies the advantages of "general text feature extraction". It uses comment samples outside the entity and unlabeled samples inside the entity to help annotate samples in the entity and learn more about common text and entity features.

We carried out multi-task improvement on the basis of the original single model, and constructed a related task two at the same time, using the hard parameter sharing method to train the model together. This effectively increases the data size of our training model. Since all tasks have varying degrees of noise, when training the model, our goal is to learn a good representation of the original task, ideally ignoring the noise associated with the data and having good generalization. Because different tasks have different noise patterns, learning the model of two tasks at the same time can learn more general characterizations. Learning only task one is possible to overfit, while joint learning tasks one and task two

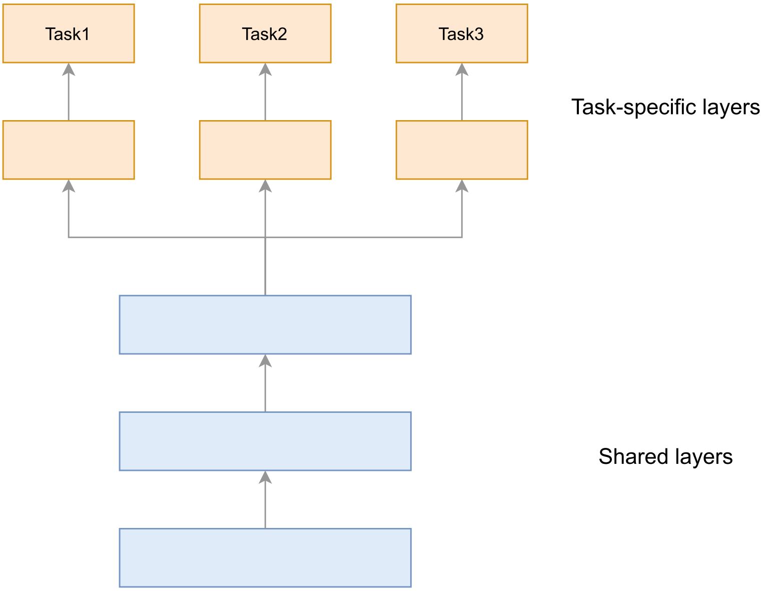
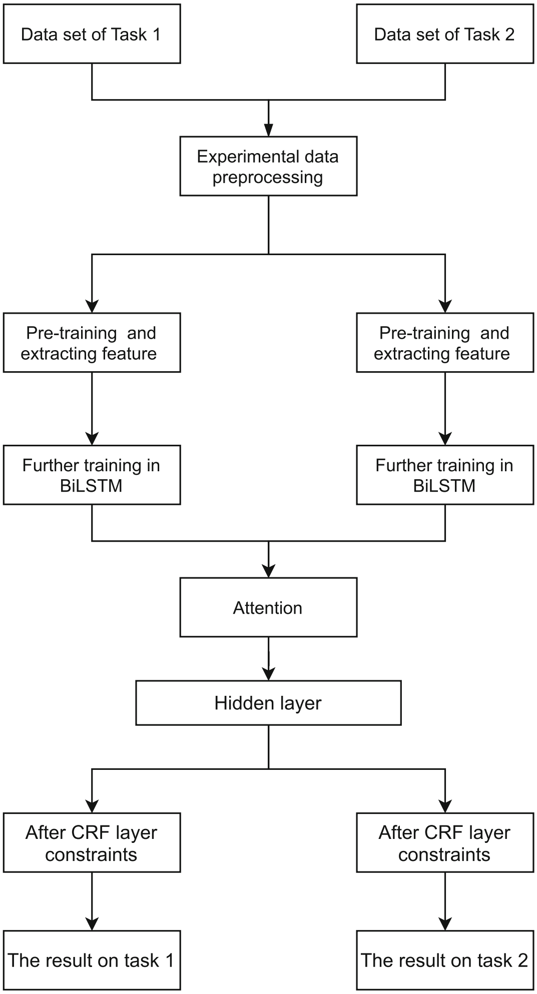


Fig. 6 Structure of hard parameter sharing mode

Fig. 7 Process of multi-task model


Data set of Task 1
Data set of Task 2
Experimental data preprocessing
Pre-training and extracting feature
Pre-training and extracting feature
Further training in BiLSTM
Further training in BiLSTM
Attention
Hidden layer
After CRF layer constraints
After CRF layer constraints
The result on task 1
The result on task 2

enables the model to obtain better characterization through the average noise pattern. The specific business process is shown in Fig. 7.

The joint learning of NER tasks in different entities mainly embodies the advantages of "general text feature extraction". It uses annotated samples outside the entity and unlabeled samples inside the entity to help annotate samples in the entity and learn more general text features and entity features.

The first model trains the auxiliary task (NER tag in our example) and then connects the second main task model (NER tag in our example) after feature extraction. By using auxiliary tasks to annotate samples, the main task can learn more common text features and entity features, to complete the annotation task better.

The word vector is input into BiLSTM for feature extraction, and then the acquired features of the secondary task are input into the hidden layer of the main task through a self-attention layer. Get the lexical features and input the final result into CRF. The output of the hidden layer is taken as the input of the CRF layer, and the final prediction result is obtained by

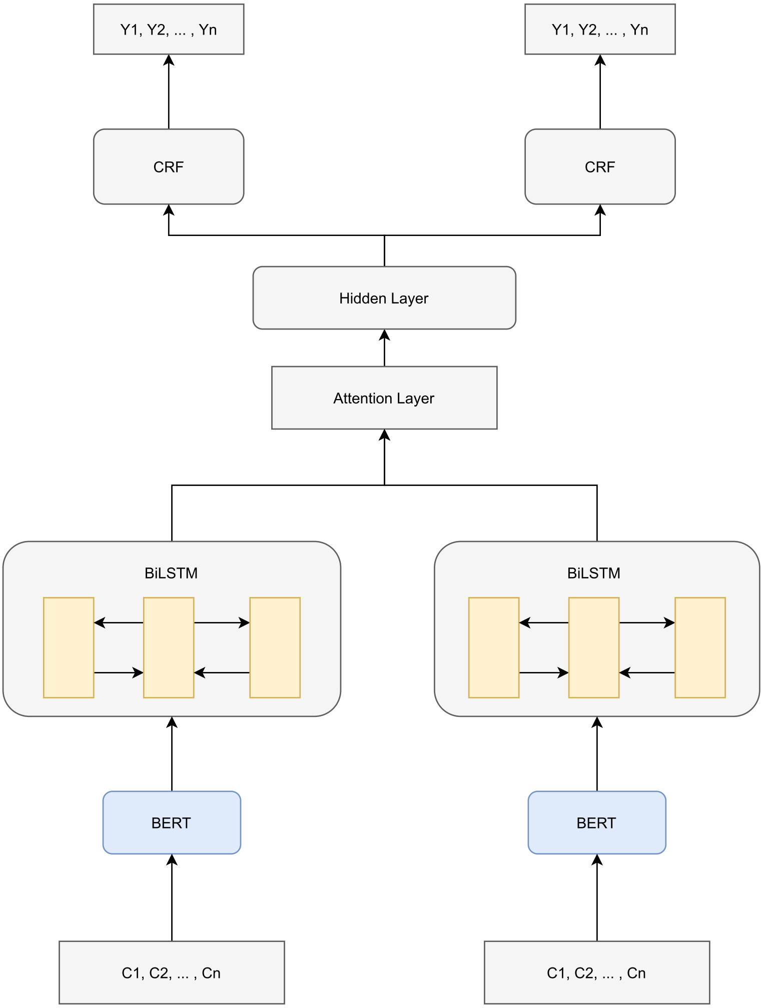


Fig. 8 The processing procedure

learning the sequence correlation information between tags. This processing procedure is shown in the Fig. 8.

The ultimate goal is to annotate the text entities in the entity, so we need to penalize the samples that are too different from the target entity. The semi-supervised learning of unlabeled samples in the field, because it directly uses model prediction to make a real label, needs to penalize the samples with low prediction confidence. Here, the author uses the optimal prediction and the percentage of relative suboptimal prediction to make $confid\,(x)$, The confidence level is dynamic, so it is necessary to predict the unlabeled data at each iteration, and then get the $confid\,(x)$.

$$confid\,(x) = \frac{\widetilde{y}\,(x) - y_{2nd}\,(x)}{\widetilde{y}\,(x)}. \tag{2}$$

The whole model framework is the joint training of labeled / unlabeled samples in the entity and labeled samples outside the entity. When the above similarity and confidence are used to adjust each iteration training, the learning rate l_r formula of different samples is expressed as

$$l_r = l_{r0} \ \times \ weight\,(x, t)\,. \tag{3}$$

$$weight\,(x, t) = \begin{cases} 1.0 & x\ in\ the\ field \\ func\,(x, IN) & x\ outside\ the\ field \\ confid\,(x, t) & x\ no\ label \end{cases} \tag{4}$$

First of all, the processed training data is featured by BERT to obtain the word vector. For any sequence, the word breaker text sequence is obtained first by word breaker processing, and then Mask on part of the word breaker sequence, unlike the Mask marker of the general BERT, the full word Musk is used for the Chinese text, the beginning of the sequence is marked with a special marker, and the sentence is separated by a marker, SEP. This results in Embedding, which consists of three parts: Token Embedding, Segment Embedding, and EmbeddIng, at which point the output of each word of the sequence vector is fed into a bidirectional Transformer for feature extraction, and finally the sequence vector $(x_1, x_1, ..., x_n)$ containing context semantic features is obtained.

Then, the sequence vector obtained by the two datasets after characteristic extraction is entered into BiLSTM, through the network cell state information forgetting and memory of new information so that the calculation of subsequent moments useful information can be transmitted, and useless information is discarded, and at each time step will output the hidden state, wherein forgetting, memory and output by the last moment of the hidden state and the current input calculated by the forgotten door, memory door, output door to control.

The first BiLSTM model training task one, and then connects the second model training task two for functional extraction. Let the training results of the two tasks be entered into the same hidden layer through the attention layer, and the final results are entered into the CRF, outputting the results separately. Because of the differences between the two datasets, we need to recognize the similarity between the main task and auxiliary task materials, and we use different learning rates for different phrase sentences. The learning rate is automatically adjusted by the similarity function. The learning rate of sentences is calculated as follows,

$$\alpha\,(x) = \alpha \cdot fun\,(x, IN)\,. \tag{5}$$

$$fun\,(x, IN) = \frac{1}{C}\frac{< v_x, v_{IN} >^d}{||v_x||^d \cdot ||v_{IN}||^d}. \tag{6}$$

where the α is a fixed learning rate, $fun\,(x, IN)$ represents the similarity between sentence x and the main task corpus IN, which ranges from 0 to 1.

By obtaining different learning rate results, the two hidden layer results, through the self-attention mechanism network, and combined with the learning rate to make a full connection. Finally, the output of the hidden layer is used as the input of the CRF layer, and the optimal label sequence is output by the CRF layer constraint.

6 Performance Evaluation

6.1 Datasets

In this work, we use the corpora provided by Microsoft Research Asia (MSRA) and the People-daily dataset to conduct the experiments. The MSRA and People-daily contain three entity types: Person (PER), Organization (ORG), and Location (LOC). The detail statistics of the corpora are summarized in Table 2.

We observe from Fig. 9 that most of the sentence length of the dataset is distributed below 150, and the maximum length of the sentence can be set to 150 for training when the data is preprocessed.

6.2 Experimental Settings

We first train a single-task model for each of the datasets, and then train in multiple MTL models, and compare the single-task performance of the training dataset with the performance of the multi-task model.

In the model training process, NER is executed as a sequential labeling task, and BIO labeling is used. The labels used are Begin-named entity, In-named entity, and Out. The named entities are different according to the entities that need to be marked in the above dataset. If a named entity is described by only one word, it will be marked as a Begin-named entity. When the named entity is described by multiple words, the beginning word will be marked with Begin-named entity, and other words will be marked as In-named entity, other words that are not named entities are marked as Out. Each dataset will be divided into a

Table 2 The statistics of datasets

Corpus	Train set	Test set	Dev set
$MSRA$	42000	3442	3000
$People - daily$	20865	4636	2318
$CLUENER$2020	10748	0	1343

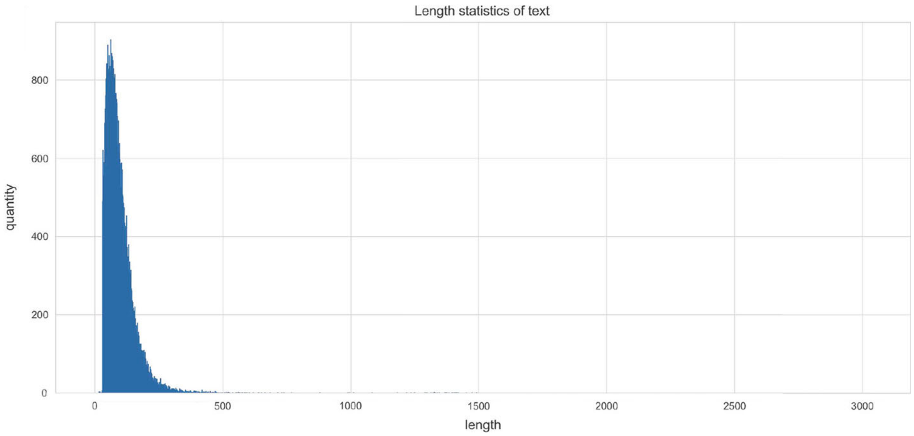


Fig. 9 The length of the text in the datasets is separate

Table 3 The detail hyper-parameters

Parameters	Values
$Optimizer$	$Adam$
$batch\_size$	32
$epoch\_size$	50
$warmup\_ratio$	0.1
$early\_stop\_ratio$	1
$cell\_type$	$lstm$
$cell\_size$	1
$rnn\_activation$	$relu$
max_seq_len	150
$label\_size$	10

training set, validation set, and test set. The model fits the data samples through the training set, adjusts the hyper-parameters of the model through the validation set, and evaluates the generalization ability of the final model on the test set.

We use the workpiece tokenizer of BERT for word breaker, relying on Google's vocab file for pre-trained BERT models. We train the model using an Adam optimizer of Tensorflow with an initial learning rate of 0.001, and the network is fine-tuned by back-propagating. For the vanishing gradient and over-fitting problems, we employ the embedding dropout method with a probability of 0.1. We control the maximum length of the sentence to be 150 and the length of the label to be 10. Otherwise, we would pad the shorter sequences, truncate the longer parts. The detail hyper-parameters are listed in Table 3.

We evaluated the accuracy of the model based on the F1 score, namely Precision (p), Recall (R), and F1 score ($F1$). Three result types were examined: the results corresponding to the positive samples correctly predicted by the model (TP), the positive samples incorrectly predicted by the model (TN), and the results corresponding to the negative samples incorrectly predicted by the model (FN). Precision (P), Recall (R) and F1 score (f) are calculated as follows.

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = 2 * \frac{P * R}{P + R} \tag{9}$$

6.3 The Evaluation Results

In addition to the single-task model and the multi-task model, two groups of comparative experiments were conducted to verify the improvement of our proposed model. We first compare our proposed strategy with existing models: BiLSTM-CRF [11], BERT-CRF, and BERT-BiLSTM-CRF [28]. The experimental results of MSRA and People's Daily data are shown in Figs. 10 and 11.

From the Fig. 10, we can conclude that our proposed model outperforms BiLSTM-CRF, BERT-CRF, and BERT-BiLSTM-CRF by (13.11%, 0.11%, 0.55%) in term of the Recall. Our proposed model is also better than BiLSTM-CRF by 10.77%, BERT-CRF by 0.48%, BERT-

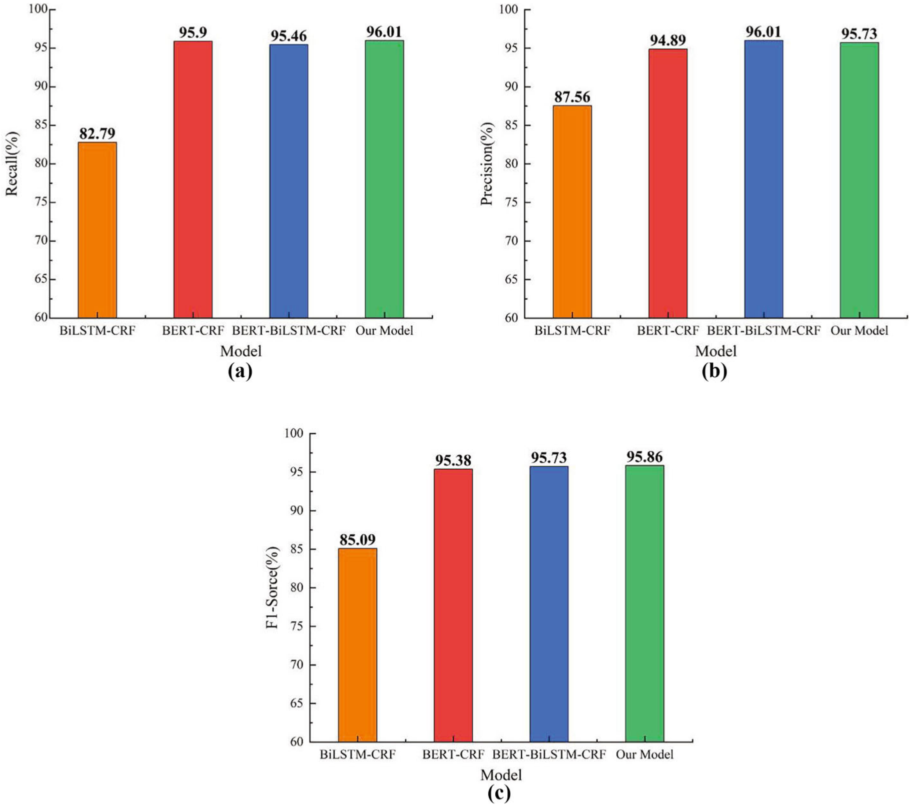


Fig. 10 The experimental results of MSRA dataset

BiLSTM-CRF by 0.13%, respectively, in term of the F1 score. For namely Precision metric, our proposed model is inferior to BERT-BiLSTM-CRF by 0.28%, better than BiLSTM-CRF by 8.17%, BERT-CRF by 0.84%, respectively.

For the Fig. 11, we can also conclude that our proposed model outperforms BiLSTM-CRF, BERT-CRF, and BERT-BiLSTM-CRF by (17.1%, 3.53%, 3.06%) in term of the Recall. Our proposed model is also better than BiLSTM-CRF by 11.82%, BERT-CRF by 3.15%, and BERT-BiLSTM-CRF by 3.76%, in term of the Precision. For namely F1 score metric, our proposed model is better than BiLSTM-CRF by 14.54%, BERT-CRF by 3.34%, BERT-BiLSTM-CRF by 3.41%, respectively.

Over all, we can obvious discovery that the models with BERT module outperforms BiLSTM-CRF in term of the Recall, Precision, and F1 score. Compared with the BiLSTM-CRF model, using Word2vec to generate static word vectors and introducing the Bert model for word vector preprocessing has achieved satisfactory results in various indicators of the experiment. From the experimental results, it is obvious that after the introduction of the Bert model, the single task model composed of Bert-BiLSTM- CRF has increased by 9.27% and 10.13%, respectively, compared with the BiLSTM-CRF model in the MSRA and people's daily datasets. Experiments show that the model can fully extract the characteristics of character level, word level, semantic level and even the relationship between sentences, so that the pre trained word vector can better represent the syntactic and semantic information

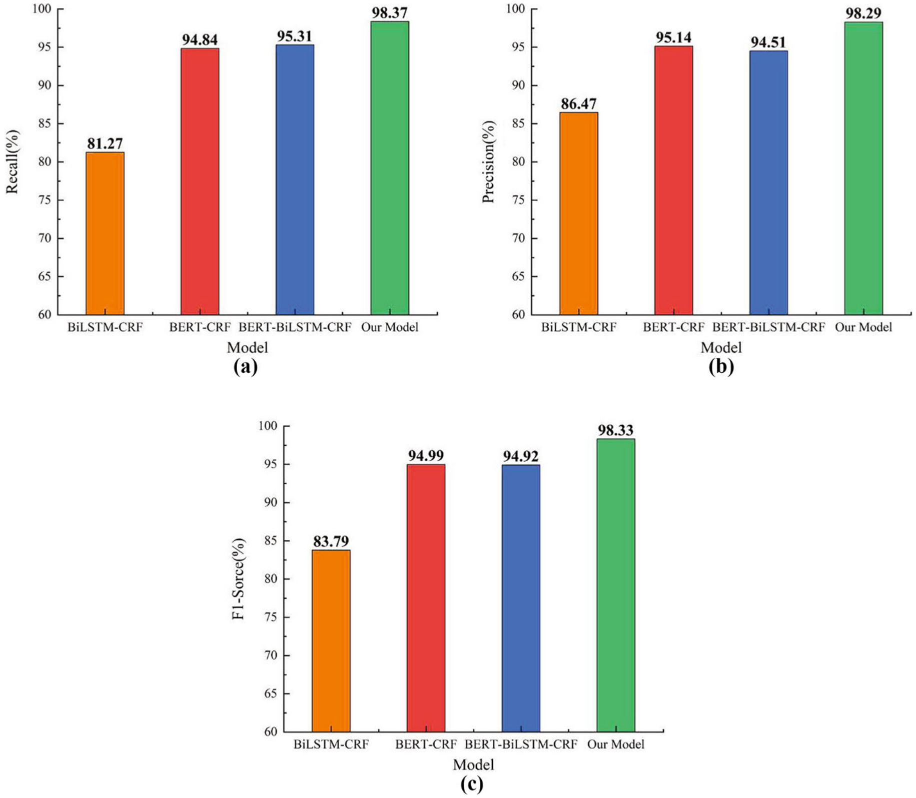


Fig. 11 The experimental results of People-daily dataset

in different contexts, enhance the generalization ability of the model and improve the performance of entity recognition. Compared with the single task model, the performance of the multi-task model is improved by 0.13% on the MSRA dataset, while the performance of the multi-task model is significantly improved by 3.41% when the people's daily dataset is small. For datasets with small amount of data, multi-task model is more obvious. And our proposed multi-task model has good performance and improved the accuracy, recall rate compared with BiLSTM-CRF, BERT-CRF, and BERT-BiLSTM-CRF.

In the second experiments, we compare our proposed strategy with existing models: CNN-BiLSTM-CRF [25], LATTICE-LSTM-CRF [26], Mixed semantic [27], and BERT-BiLSTM-CRF [28]. The experimental results are shown in Table 4. From Table 4, we can conclude that our proposed model outperforms CNN-BiLSTM-CRF, LATTICE-LSTM-CRF, Mixed semantic, and BERT-BiLSTM-CRF by (5.6%, 3.3%, 1.2%, 2.0%) in term of the Recall. Our proposed model is also better than CNN-BiLSTM-CRF by 4.9%, LATTICE-LSTM-CRF by 2.7%, Mixed semantic by 0.5%, BERT-BiLSTM-CRF by 1.7%, respectively, in term of the F1 score. For namely Precision metric, our proposed model is inferior to Mixed semantic by 0.2%, but better than CNN-BiLSTM-CRF by 4.2%, LATTICE-LSTM-CRF by 2.2%, BERT-BiLSTM-CRF by 1.4%, respectively. In a word, our proposed multi-task model has good performance and improves the accuracy, recall rate compared with other models.

Table 4 The comparison with other models on MSRA dataset (unit: %)

Model	P	R	$F1$
CNN-BiLSTM-CRF	91.63	90.56	91.09
LATTICE-LSTM-CRF	93.57	92.79	93.18
Mixed semantic	**95.92**	94.80	95.36
BERT-BiLSTM-CRF	94.35	94.07	94.21
Our model	95.73	**96.01**	**95.86**

Bold values indicate the optimal value in such evaluation criteria

Table 5 The experimental results of CLUNER2020 dataset (unit: %)

Entity	P	R	$F1$
Name	75.16	74.19	74.68
Address	57.43	46.65	51.48
Movie	76.43	70.86	73.54
Position	78.83	71.36	74.91
Organization	77.38	77.38	77.38
Company	73.96	75.13	74.54
Scene	60.66	53.11	56.63
Government	74.23	78.14	76.13
Book	78.95	68.18	73.17
Game	76.28	80.68	78.42

In order to further explore the applicability of the multi-task model based on the above experiments, we conducted multi-task learning experiments on the CLUNER2020 dataset. The original two experimental datasets in this paper contain only three entity tags (ORG, PER and LOC). The words and entity labels in the data set are expressed in one-to-one form. The CLUNER2020 dataset is a NER fine-grained task and contains ten entity tags (name, address, movie, organization, company, scene, government, book, game). The experimental results of CLUNER2020 dataset are shown in Table 5.

We can see the recall rate, accuracy rate and F1 score of multi-task model on the CLUNER2020 data set about the above ten entities. Compared with the multi task learning results of MASR and People-daily datasets, the efficiency of this model on this data set is not ideal. In order to further compare the experimental results, the experimental results are compared with the two types of models on the CLUNER2020 dataset [33]. The experimental results are shown on the Table 6.

From the Table 6, we can conclude that our proposed model outperforms BiLSTM-CRF by 1.61% in term of the F1-score. However, our proposed model is inferior than Single task model based on Bert base by 0.28%. Through The comparison with other models on the CLUNER2020 dataset, we found that the final learning effect is only slightly better than the baseline model and much worse than the single task model based on Bert.

The experimental effect of multi-task model on this task is not ideal, which may be due to the following reasons:

- First, the label difference between the two datasets of multi-task learning is too large. MASR dataset adopts three entities marked by BIO, and CLUNER2020 dataset adopts ten entities marked by BIO. The entity tags in the CLUNER2020 dataset may be stacked in the Mars dataset, such as government and LOC tags.

Table 6 Compare F1 sorce of the other models on CLUNER2020 dataset (unit: %)

Entity	BiLSTM CRF	BERT base	Our model
Name	74.04	88.75	74.68
Address	45.50	60.89	51.48
Movie	78.97	85.82	73.54
Position	70.16	78.89	74.91
Organization	75.96	79.43	77.38
Company	72.27	81.42	74.54
Scene	52.42	65.10	56.63
Government	77.25	87.03	76.13
Book	67.20	73.68	73.17
Game	85.27	86.42	78.42
Overall@Macro	70.00	78.82	71.61

- Second, sometimes there are different labels for the expression of the same entity. The person's name is marked as org in the original dataset and name in the CLUNER2020 dataset. At this time, multi-task learning may not play the role of supplementary learning, but will produce noise and affect the judgment of the model on the original entity type.
- Finally, the multi-task learning mode in this paper uses the information contained in the relevant task training to improve the generalization ability. For the newly added CLUNER2020 dataset, the general text features cannot be extracted in the parameter sharing layer model.

7 Conclusion

In this paper, we propose a multi-task BERT-BiLSTM-AM-CRF intelligent processing model, which can be beneficial to text mining tasks on some Chinese datasets. First, we propose a BERT-BiLSTM-CRF single task model, where the dynamic word vector combined with context information is extracted by BERT. And, after further training by BiLSTM module, the result is input into CRF layer for decoding. The CRF can classify and extract the obtained observation annotation sequence to obtain the final result. Then, a multi-task learning model is constructed. This model extracts the word vector through BERT and input it into BiLSTM for feature learning. Through the attention mechanism network, the model can learn together on two Chinese datasets, and finally get the training results through CRF layer constraints.

Compared with single-task learning, we observed that multi-task learning has significant improvement in some datasets. Although the performance of some tasks decreased after adding some auxiliary datasets, the performance of most tasks improved significantly. This is a promising result, which shows the potential of MTL in NER. However, there are still many limitations to multi-task. For example, it is difficult to predict when these multitasking learning models will benefit. In addition, we can see the deficiency of multi-task learning from the final experiment. We make different entity annotation rules for the two datasets and put them into the model for multi-task learning. The experimental results are worse than the single task model. The application of multi-task learning on NER is waiting for these problems to be solved further and bring greater practical value.

Acknowledgements This research was partially funded by National Natural Science Foundation of China (Grant No. 61972146), Hunan Provincial Natural Science Foundation of China (Grant No. 2020JJ4376).

References

1. Xie X, Fu Y, Jin H, Zhao Y, Cao W (2020) A novel text mining approach for scholar information extraction from web content in chinese. Futur Gener Comput Syst 111:859–872
2. Xiang L, Sun X, Luo G, Xia B (2014) Linguistic steganalysis using the features derived from synonym frequency. Multimed Tools Appl 71(3):1893–1911
3. Sun C, Yang Z, Wang L, Zhang Y, Lin H, Wang J (2021) Biomedical named entity recognition using bert in the machine reading comprehension framework. J. Biomedical Informatics 118:103799
4. Zhai Z, Nguyen DQ, Akhondi S, Thorne C, Verspoor K (2019) Improving Chemical Named Entity Recognition in Patents with Contextualized Word Embeddings. Proceedings of the 18th BioNLP Workshop and Shared Task
5. Ivan L, Nicolas P, Xavier T (2020) Terminologies augmented recurrent neural network model for clinical named entity recognition. J. Biomedical Informatics 102:103356
6. Zhou S, Tan B (2020) Electrocardiogram soft computing using hybrid deep learning cnn-elm. Appl Soft Comput 86:105778
7. He S, Li Z, Tang Y, Liao Z, Li F, Lim S (2020) Parameters compressing in deep learning. Cmc-computers Materials & Continua 62(1):321–336
8. Habibi M, Weber L, Neves M, Wiegandt DL, Leser U (2017) Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics 33(14):37–48
9. Lample G, Ballesteros M, Subraman S (2016) Neural architectures for namedentity recognition. Proceedings of NAACL-HLT, 260–270
10. Ma X, Hovy EK (2016) End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics
11. Chen Y, Zhou C, Li T, Wu H, Zhao X, Ye K, Liao J (2019) Named entity recognition from chinese adverse drug event reports with lexical feature based bilstm-crf and tri-training. J. Biomedical Informatics 96:103252
12. Li Z, Li Q, Zou X, Ren J (2021) Causality extraction based on self-attentive bilstm-crf with transferred embeddings. Neurocomputing 423:207–219
13. Putra FM, Retno K, Adi W (2021) Sentiment analysis using word2vec and long short-term memory (lstm) for indonesian hotel reviews. Procedia Comp Sci 179:728–735
14. Quang-Thai H, Trinh-Trung-Duong N, Nguyen QKL, Yu-Yen O (2021) Fad-bert: Improved prediction of fad binding sites using pre-training of deep bidirectional transformers. Comput Biol Med 131:104258
15. Tang X, Cao W, Tang H, Deng T, Mei J, Liu Y, Shi C, Xia M, Zeng Z (2022) Cost-efficient workflow scheduling algorithm for applications with deadline constraint on heterogeneous clouds. IEEE Trans. Parallel Distrib Syst 33(9):2079–2092
16. Tang X, Shi C, Deng T, Wu Z, Yang L (2021) Parallel random matrix particle swarm optimization scheduling algorithms with budget constraints on cloud computing systems. Appl Soft Comput 113:107914
17. Fukuda K, Tsunoda T, Tamura A et al (1998) Toward information extraction: identifying protein names from biological papers. Pac Symp Biocomput 707(18):707–718
18. Hanisch D, Fundel K, Mevissen HT et al (2005) Prominer: rule-based protein and gene entity recognition. BMC bioinformatics 6(1):14
19. Lee KJ, Hwang YS, Kim S et al (2004) Biomedical named entity recognition using two-phase model based on svms. J. Biomedical Informatics 37(6):436–447
20. Satyajit N, Justin D (2022) Factored latent-dynamic conditional random fields for single and multi-label sequence modeling. Pattern Recogn 122:108236
21. Chen H, Sun F, Yuan J, Huan Y (2021) Mirrored conditional random field model for object recognition in indoor environments. Inf Sci 551:291–303
22. Liu X, Zhou Y, Wang Z (2021) Deep neural network-based recognition of entities in chinese online medical inquiry texts. Futur Gener Comput Syst 114:581–604
23. De Oliveira DM, Laender AHF, Veloso A et al (2013) FS-NER: A lightweight filter-stream approach to named entity recognition on twitter data. Proceedings of the 22nd International Conference on World Wide Web
24. Liu P, Guo Y, Wang F, Li G (2022) Chinese named entity recognition: The state of the art. Neurocomputing 473:37–53

25. Jia Y, Xu X (2018) Chinese named entity recognition based on CNN-BiLSTM-CRF. IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)
26. Zhao S, Cai Z, Chen H, Wang Y, Liu F, Liu A (2019) Adversarial training based lattice lstm for chinese clinical named entity recognition. J. Biomedical Informatics 99:103290
27. Chang N, Zhong J, Li Q, Zhu J (2020) A mixed semantic features model for chinese ner with characters and words. In: ECIR 2020: Advances in Information Retrieval, pp. 356–368. Springer, Heidelberg
28. Dai Z, Wang X, Ni P, Li Y, Li G, Bai X (2019) Named entity recognition using BERT BiLSTM CRF for Chinese electronic health records. The 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)
29. Murugesan G, Abdulkadhar S, Bhasuran B, Natarajan J (2017) Bcc-ner: bidirectional, contextual clues named entity tagger for gene/protein mention recognition. J. Bioinform Sys. Biology 7:1–8
30. Cheng P, Dai J, Liu J (2022) Catvrnn: Generating category texts via multi-task learning. Knowl-Based Syst 244:108491
31. Xu K, Zhou Z, Gong T, Hao T, Liu W (2018) Sblc: a hybrid model for disease named entity recognition based on semantic bidirectional lstms and conditional random fields. BMC Med Inform Decis Mak 18:114
32. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing
33. Xu L, Dong Q, Yu C, Tian Y, Liu W, Li L, Zhang X (2020) Cluener2020: Fine-grained name entity recognition for chinese. arXiv preprint arXiv:2001.04351

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.