A Collaborative Optimization-Guided Entity Extraction Scheme

Qiaojuan Peng[1,2,3], Xiong Luo[1,2,3(✉)], Hailun Shen[4], Ziyang Huang[4], and Maojian Chen[1,2,3]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
xluo@ustb.edu.cn
[2] Shunde Graduate School, University of Science and Technology Beijing, Foshan 528399, China
[3] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China
[4] Ouyeel Co., Ltd., Shanghai 201999, China

Abstract. Entity extraction as one of the most basic tasks in achieving information extraction and retrieval, has always been an important research area in natural language processing. Considering that most of the traditional entity extraction methods need to manually adjust their hyperparameters, it takes a lot of time and is easy to fall into local optimality. To avoid such limitations, this paper proposes a novel scheme to extract named entities, where the model hyperparameters are automatically adjusted to improve the performance of entity extraction. Here, the proposed scheme is composed of bi-directional encoder representation from transformers (BERT) and conditional random field (CRF). Specifically, through the fusion of collaborative computing paradigm, particle swarm optimization (PSO) algorithm is utilized in this paper to search for the best value of hyperparameters automatically in a cooperative way. The experimental results on two public datasets and a steel inquiry dataset verify that our proposed scheme can effectively improve the performance of entity extraction.

Keywords: Entity extraction · Particle swarm optimization (PSO) · Bi-directional encoder representation from transformers (BERT) · Collaborative optimization

1 Introduction

Currently, the automatic extraction of information from unstructured data has attracted extensive attention. As a basic task in information extraction and retrieval, entity extraction identifies entities with specific meanings, such as location, person, proper noun, and some others from the text [24]. With the development of natural language processing (NLP) technologies, text semantic

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 407, pp. 190–205, 2021.
https://doi.org/10.1007/978-3-030-92638-0_12

knowledge is becoming more and more important. Recently emerging research fields such as automatic question answering, intelligent search and semantic analysis, require rich semantic knowledge as support. As the most important semantic knowledge in text, entity extraction has always been an important area. However, there are three difficulties in this field. First, the named entity itself has the characteristics of complexity, diversity and randomness, then it is difficult to accurately define and classify its entity type. Second, since it may lack large-scale knowledge databases like Wikipedia in practically industrial applications, it is challenging to obtain a large amount of high-quality annotation data. Third, due to the domain characteristics of different applications, there are many out of vocabulary (OOV) words, i.e., new words, which takes a lot of time and effort to recognize them manually.

In order to cope with the above challenges, some unsupervised and supervised machine learning algorithms are used to label large-scale data automatically [19,25,33,34]. To further improve the performance of achieving this task, we develop a novel scheme through the combination of popular deep learning model and typical evolutionary algorithm. During the implementation process, many important hyperparameters in those algorithms need to be adjusted to guarantee that the satisfactory computing performance of learning model could be achieved. Then, the adjustment that previously were based on painstakingly handcrafted operations, can now be made using intelligent strategies.

Generally, collaborative computing is an efficient paradigm in which different groups can work together for a certain task in a coordinated manner. It indicates that groups in a dispersed state cooperate with each other to achieve a task together, while more effectively promoting the collaboration between social groups and greatly improving the working quality of the group. In this field, particle swarm optimization (PSO) is a typical evolutionary algorithm [16], which aims to address optimization problems with the help of collaboration and information sharing between particles. Through the fusion of it, in our developed scheme PSO algorithm is utilized to intelligently search for the best value of hyperparameters in a cooperative way.

More specifically, we propose a novel scheme to achieve entity extraction using an advanced architecture, which is composed of bi-directional encoder representation from transformers (BERT) and conditional random field (CRF). This scheme is called BERT-CRF-PSO. Among them, BERT uses the encoder architecture in the transformer to obtain the semantic vector and combines with the masked language model (MLM) to achieve contextual prediction [12]. The CRF [17] can effectively use past and future labels to predict the current label, so as to obtain more accurate entity prediction conditional probabilities. Meanwhile, we introduce the PSO to fine-tune the hyperparameters of the BERT model. The probability value of the entity prediction can be obtained finally. In general, the BERT and CRF are responsible for the training data, and the PSO is responsible for fine-tuning the hyperparameters. These three modules share information and collaborate with each other to complete the entity

extraction task together. Through this model, it is expected that we can improve the performance of entity extraction.

The contributions of this paper can be summarized as follows.

- Applying collaborative computing to the field of entity extraction, so as to efficiently realize entity extraction in a way of mutual cooperation and information sharing.
- In order to improve the performance of entity extraction, the PSO algorithm is used to automatically fine-tune the hyperparameters of the BERT model, so as to find the global optimal hyperparameters for a specific dataset.

The rest of this paper is organized as follows. In Sect. 2, the related work of entity extraction is simply introduced. In Sect. 3, we present the proposed scheme BERT-CRF-PSO. The relevant experimental results are shown in Sect. 4. Finally, in Sect. 5, our work is summarized.

2 Related Work

The methods of implementing entity extraction can be classified into three categories. In this section, we introduce some related algorithms about entity extraction.

2.1 The Rule and Vocabulary-Based Entity Extraction

Early, the entity extraction was achieved using a rule and vocabulary-based method. The basic idea was to select features, e.g., statistical information, keywords, and punctuation, to construct specific rules manually, and use methods, e.g., pattern and string matching, to perform for entity extraction.

Xie *et al.* proposed a method of combining manually formulated rules with heuristic ideas, and realized the automatic recognition of named entities from unstructured text [32]. Berry *et al.* used a self-training model to conduct entity extraction on the basis of unified medical language system (UMLS) [2], and the optimal result of F_1 reached 85.23%. Farmakiotou *et al.* proposed an entity extraction system based on manual vocabulary resources [14]. The rules were mainly formulated for the Greek language and included hand-made grammar and gazetteers. The system had achieved satisfactory test results for the experiments on a Greek corpus of financial news. Collins *et al.* proposed the DL-CoTrain method [9]. The basic processing of this method could be divided into three steps. Firstly, the seed rule set was pre-defined. Secondly, multiple unsupervised training were performed on the pre-defined rule set to obtain more rules. Finally, the rule set was used to extract named entities. The classification accuracy of this method for three categories of person, location and organization all exceeded 91%.

This class of methods achieved good performance when the extracted rules could accurately reflect the language characteristics in a specific field. However, the feature of those methods had also become their shortcoming. It relied too much on a specific language, domain, and text style, which might lead to poor portability of the model.

2.2 The Traditional Machine Learning-Based Entity Extraction

For those methods using traditional machine learning models, entity extraction were generally regarded as a sequence tagging task. Compared with the classification problem, the predicted label sequences in the sequence tagging task had a very strong interdependence. In other words, the currently predicted label was not only related to the current input feature, but also related to the previous predicted label. Here, in this field, some traditional machine learning were used, and they mainly included CRF, maximum entropy (ME) [28], support vector machine (SVM) [18], and hidden Markov model (HMM) [8].

Among the above four methods, CRF could provide a globally optimal label model for entity extraction. However, it took a long training time and its convergence speed was much slow. ME had the advantages of compact structure and good versatility. Its disadvantage was that explicit normalization calculations were required in the training processing, while leading to relatively large costs. HMM used the viterbi algorithm to address the problem of recognizing entities from sequence. Therefore, its training speed and recognition speed were fast, while it performed worse than ME and SVM in accuracy.

Das *et al.* used the CRF model to achieve entity extraction tasks for Indian languages [11]. The accuracy of this model for entity extraction on Bengali and Hindi were 87% and 79%, respectively. Saha *et al.* used the ME model to extract four types of entities, i.e., person, organization, location, and date, on Hindi, and the value of final F_1 was 81.52% [28]. Lin *et al.* proposed a entity extraction system on the basis of SVM model, and used the system to achieve high accuracy [18]. Chopra *et al.* used HMM for entity extraction tasks on the Treebank corpus with 25 files. In the end, they achieved the result of F_1 is 73.8% on eight types of named entities [8].

2.3 The Deep Learning-Based Entity Extraction

Recently, with the development of deep learning technologies, they were gradually popular in the field of entity extraction [6]. Compared with the above two classes of methods, the deep learning-based methods had three main advantages [30,38]:

- They could learn more complex features and potential knowledge from the original data text through a nonlinear activation function.
- They were end-to-end, which meant that it could avoid error propagation between modules in the pipeline model and could carry more complex internal designs, so as to achieve better experimental results.
- Experiments had verified that the deep learning algorithms were very suitable for processing sequence tagging problems, and did not rely on domain knowledge and feature engineering.

Chiu *et al.* proposed a two-way model combining long short-term memory (LSTM) and convolutional neural network (CNN), which could automatically capture word-level and character-level features [7]. Ma and Hovy

proposed a novel neutral network architecture combining bidirectional LSTM (BiLSTM), CNN and CRF, which would automatically benefit from word-level and character-level representations [23]. Zhao *et al.* developed a entity extraction method through the use of CNN. They took word-level embedding, dictionary features, and others as input, and used CNN to automatically extract useful features [36]. Liu *et al.* developed a neural network framework called LM-LSTM-CRF, which merged the character-aware neural language model to extract character-level knowledge. Unlike most transfer learning methods, the framework proposed by Liu *et al.* did not rely on any additional supervision and could identify new entities well [20]. More recently, adding graph neural network, attention mechanism, transfer learning and other technologies to neural network-based models was also a popular direction in the implementation of entity extraction [4,5,27,31].

3 The Proposed Scheme

The overall architecture of our scheme is shown Fig. 1, where [CLS] is a special symbol for classification output, N is the number of characters in the text, Tok_i represents the i-th character, E_i represents the i-th token embedding, T_i represents the final i-th embedding vector, and Tag_i is the final output tag sequence. Here, i is bounded within $[1, N]$.

It can be seen from this figure that, this architecture mainly consists of three modules, including BERT model, CRF layer, and PSO algorithm. Among them, the BERT model effectively achieves the word embedding, transformer encoding and pooling, and it is used to pre-train the language model to obtain the corresponding word vector. The CRF layer adds some constraints to the final predicted label to ensure that it is legal and correct. The CRF layer can automatically learn some rules during the training processing. For example, if a dataset uses the BIO labeling scheme, where "B" represents the beginning of a named entity, "I" represents the middle of a named entity, and "O" represents the non-entity. Meanwhile, the constraints learned by CRF layer are as follows:

- The first word in a sentence always starts with the label "B-" or "O-" instead of "I-".
- A pattern can be learned by "B-label1 I-label2 I-label3 $\cdots$", which means that labels 1, 2, and 3 should be the same entity category. For instance, "B-Location I-Location I-Location" is a legal label sequence. However, "B-Location I-Location I-Person" is an illegal label sequence.
- The label sequence "O I- $\cdots$" is illegal. In other words, the legal label sequence should be "O B- I-".

The PSO algorithm simulates the foraging behavior of a flock of birds. In our model, it iteratively searches for BERT hyperparameters which most suitable for a specific dataset, and continuously adjusts the hyperparameters during the search processing to achieve the best performance of the BERT-CRF model. In short, these three different modules work together to complete a task in a way of

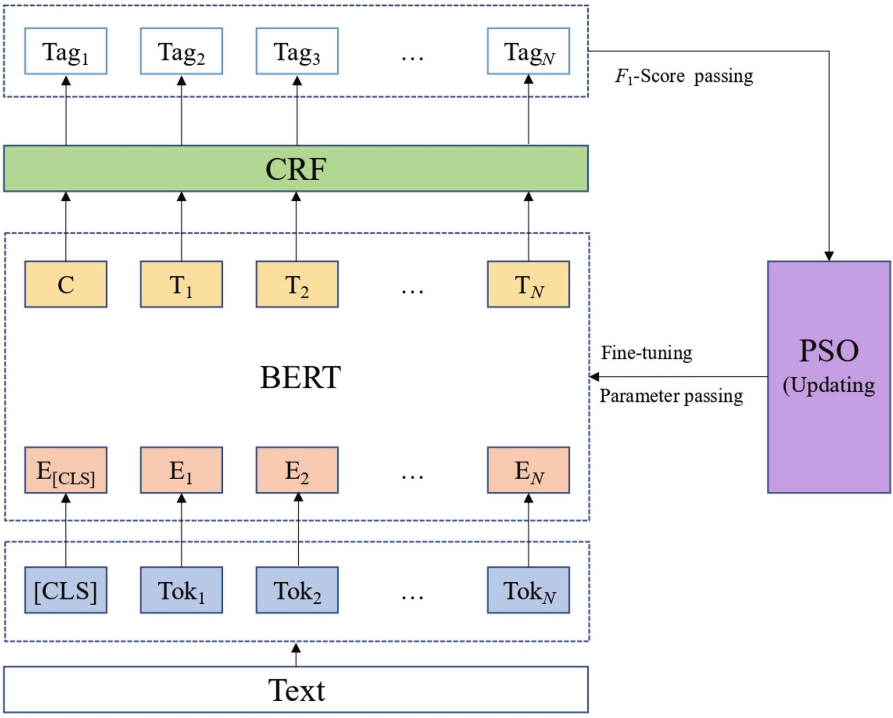

Fig. 1. The architecture of our proposed scheme.

mutual cooperation and information sharing, so as to achieve entity extraction efficiently and accurately.

3.1 The Use of BERT Model

BERT uses a multi-layer two-way transformer as a feature extractor, and uses a self-attention mechanism to make each word have global semantic information, thus capturing long-distance dependencies. In addition, BERT uses the Word-Piece method to build the vocabulary. WordPiece can be understood as splitting a word into subwords. If a word is not in the vocabulary, the tokens are split one by one in the way of subwords. If no token is found, [unknown] is directly assigned. In this way, the information of the root of the word can be effectively captured, the vocabulary capacity can be reduced, and the OOV problem can also be alleviated. BERT contains three important parts: the MLM model, the transformer model, and the next sentence prediction (NSP).

The MLM Model. BERT uses a MLM to overcome the limitation of one-way information. In this scheme, 15% of the words in the dataset are masked randomly, and then the BERT model is used to predict these masked words, which is a bit similar to the cloze task we are familiar with. In this way, the BERT model can rely more on context-related information to predict the vocabulary.

The Transformer Model. Transformer is divided into two parts: encoder and decoder. The architecture of transformer is shown in Fig. 2 [29]. Encoder is a stack of N identical layers, and each layer has two sublayers. The first layer is a multi-head self-attention mechanism, and the second layer is a fully connected feedforward neural network. After the two sub-layers, each layer is connected by residuals, and then layer standardization is performed. Finally, a N-layer encoder is formed. The decoder is also stacked by N identical layers, but the difference from the encoder layer is that the layer in the decoder is composed of a multi-head attention and "Add&Norm" inserted into the encoder layer. This method makes full use of the context and does not require bidirectional stacking such as BiLSTM.

In fact, the BERT model only uses encoder of transformer to encode the input. This is because BERT is essentially a pre-training model. It is carried out through the language model, and it is different from other specific tasks of NLP. Meanwhile, since BERT currently does not have a decoder of transformer, it also reduces a lot of unnecessary operations in its attention function.

Next Sentence Prediction. The role of NSP can be understood that as two sentences in a given text, it is to determine whether the second sentence immediately follows the first one. Some tasks require a model to understand the relationship between two sentences. However, this goal can not be achieved by only training the language model. This is the reason why BERT trains NSP.

Therefore, when BERT pre-trains the data, there is a 50% probability that the model will choose two contextual sentences A and B, and a 50% probability that it will choose two context-independent sentences A and B.

3.2 The Design of CRF Layer

In order to make the classifier perform better, when labeling data, we can consider using the labeling information of adjacent data. It is difficult for general classifiers to do this. However, CRF is particularly good at handling contextual information. Because the typical characteristic of CRF is to use a log-linear model to represent the joint probability of the feature sequence, which is convenient for people to effectively use the context label to predict the current label. CRF is a typical sequence tagging algorithm, that is, given an input sequence $\mathbf{X} = \{x_1, x_2, x_3, \cdots, x_{n-1}, x_n\}$, the target sequence $\mathbf{Y} = \{y_1, y_2, y_3, \cdots, y_{n-1}, y_n\}$ is the output of model. It has been employed in sequence tagging tasks such as word segmentation, part-of-speech tagging and entity extraction [10,15,22]. The structure of CRF is shown in Fig. 3 [17].

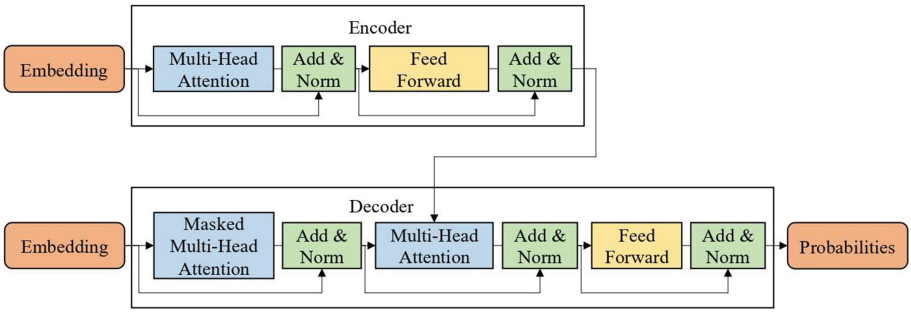

Fig. 2. The architecture of transformer.

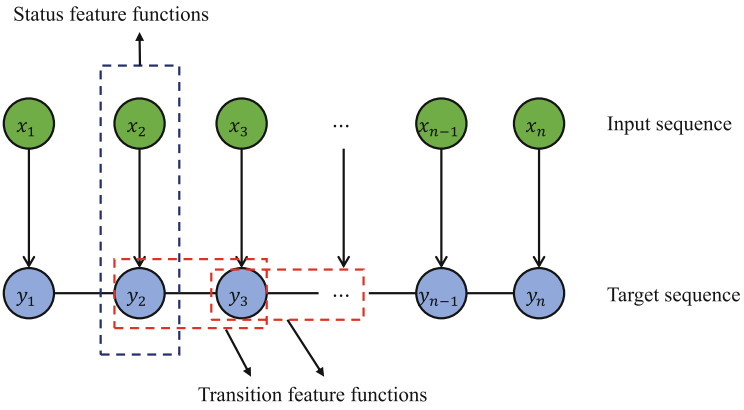

Fig. 3. The structure of CRF.

Let $\mathbf{X} = \{x_1, x_2, x_3, \cdots, x_{n-1}, x_n\}$ be the input sequence of sentence whose length is n, and let $\mathbf{Y} = \{y_1, y_2, y_3, \cdots, y_{n-1}, y_n\}$ be the corresponding output target sequence. Assuming that $P(\mathbf{Y}|\mathbf{X})$ is a linear chain conditional probability, it is shown in (1).

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z}\exp\left(\sum_{i=1}^{n-1}\sum_{k=1}^{K}\lambda_k t_k\left(y_{i+1}, y_i, \mathbf{X}, i\right) + \sum_{i=1}^{n}\sum_{l=1}^{L}u_l s_l\left(y_i, \mathbf{X}, i\right)\right), \quad (1)$$

where K is the total number of transition feature functions defined at nodes, L is the total number of status feature functions defined at nodes, i is the position of the current node in sequence, λ_k and u_l are weights, and Z is a normalized factor. Moreover, t_k and s_l are transition feature functions and status feature functions, respectively.

3.3 The PSO-Based Collaborative Optimization

PSO is a population-based random optimization algorithm. The core idea of this algorithm is to simulate the foraging behavior of a flock of birds. In the process of foraging, the birds continuously communicate and share their positions and information with each other, so that other birds know their positions. Through the collaboration between the group, each bird judges whether the position it finds is the optimal solution (the closest food source), and at the same time, they also transmit the information of the optimal solution to the entire bird group. In each iteration process, each bird adjusts its position according to the optimal solution found by itself and the optimal solution of the population to ensure that the entire flock of birds can finally gather around the food source, that is, the optimal solution is found. The advantage of PSO is that it is easy to implement, fast to converge, and does not need to adjust a large number of parameters. At present, it has been widely used in neural networks, data mining, image processing and other fields [1,3,26,37].

The best value of BERT model hyperparameters will help the model perform better on a specific dataset, which is essentially an optimization problem. Here, PSO is particularly suitable for dealing with optimization problems. In PSO, the position of birds in the searching space represents the solution space of the problem, and these birds are also as "particles". Every particle has three basic attributes, including position, speed, and fitness. The position and speed determine the direction and distance of the particles, and the fitness is determined by an optimization function. In this paper, the position refers to the hyperparameters of the BERT model, the speed represents the direction of the change of hyperparameters, the optimization function is the BERT-CRF model, and the fitness can be measured by F_1-Score. Actually, the metrics used for evaluating the effectiveness of model usually include precision, recall, and F_1-score. However, precision and recall are a pair of contradictory measures. Generally, when the precision is high, the recall tends to be low, and when the precision is low, the recall tends to be high. Since F_1-score is a weighted average of precision and recall, it can take into account the precision and recall of classification model at the same time. Hence, we choose F_1-score to measure fitness in our scheme. Each particle has a memory function, which can save all historical solutions. While searching, they can follow the best optimal particle in a certain iteration in the population, and find the next solution in accordance with the solution of the best optimal particle.

PSO initializes a group of particles, so that they have random initial positions and speeds. In the iterative process, each particle updates its attributes by tracking two "extreme", i.e., the historically optimal solution of particle itself, called the individual extreme point (using $\overline{\text{pbest}}$ to represent its position), the historically optimal solution of entire population, called the global extreme point (using $\overline{\text{gbest}}$ to represent its position). In fact, $\overline{\text{pbest}}$ can be regarded as the flying experience of the particle itself (individual cognition), and $\overline{\text{gbest}}$ can be regarded as the flying experience of the particle's companion (social behavior). By finding these two optimal solutions, the particles update

their speeds and positions according to (2) and (3). Assuming that the information of the i-th particle is represented by a D-dimensional vector, where D is the number of hyperparameters required to adjust, the position can be expressed as $\mathbf{X}_i = (x_{i1}, x_{i2}, \cdots, x_{id}, \cdots, x_{iD})^\top$, and the speed can be expressed as $\mathbf{V}_i = (v_{i1}, v_{i2}, \cdots, v_{id}, \cdots, v_{iD})^\top$. Then, we have:

$$v_{id}^{k+1} = w \times v_{id}^{k} + c_1 \times \text{rand}_1^k \times (\overline{\text{pbest}}_{id}^{k} - x_{id}^{k}) + c_2 \times \text{rand}_2^k \times (\overline{\text{gbest}}_{d}^{k} - x_{id}^{k}), \quad (2)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}, \quad (3)$$

where $v_{id}^k (1 \leqslant d \leqslant D)$ represents the velocity of the d-th hyperparameter of the i-th particle during the k-th iteration. The larger the inertia factor w, the stronger its global search ability, and the smaller the inertia factor w, the stronger its local search ability. So the introduction of inertia factor w can adjust the global and local searching capabilities of algorithm. Moreover, c_1 and c_2 are learning factors used to adjust the maximum step length during the particle flight. Appropriate c_1 and c_2 can speed up the convergence of algorithm and make the results avoid local minimum. Moreover, rand_1 and $\text{rand}_2 \in [0, 1]$ are random values. In addition, x_{id}^k represents the position of the d-th dimension of the i-th particle during the k-th iteration. Finally, $\overline{\text{pbest}}_{id}$ is the position of the individual extreme point of the i-th particle during the d-th dimension, and $\overline{\text{gbest}}_d$ is the position of the global extreme point of entire population during the d-th dimension. In addition, in order to prevent the particles from gradually moving away from the searching space, the position and velocity of each dimension of the particles will be limited to an interval [13].

Then, the parameters optimization via PSO is shown in Algorithm 1.

4 Experiments

4.1 Experimental Dataset

The experiment in this paper uses a total of three datasets. The first type is the boson dataset[1], which includes six types of entities: time, location, person, organization, company, and product. The second type is the People's Daily dataset[2], including three named entities of organization, institution and person. The last type is the inquiry data of steel industry. It possesses 744 data including eight entity categories, such as grade, variety, specification, place of production, thickness, weight, surface structure, and surface treatment. It is noted that both the boson and the People's Daily datasets can be obtained on the Internet. The steel dataset is the historical inquiry information of customers on the steel e-commerce platform. Considering the consistency of different datasets, we select 1,000 corpora from the boson dataset and 1,000 corpora from the People's Daily dataset in the experiment.

[1] http://static.bosonnlp.com/dev/resource.
[2] http://www.ling.lancs.ac.uk/corplang/pdcorpus/pdcorpus.html.

Algorithm 1. The implementation process of PSO

1: Set the fitness: F_1-Score;
2: Set the position vector (BERT model hyperparameters): $[x_1, x_2, \cdots, x_D]$;
3: Set the maximum number of iterations: $k_{\max}$;
4: Initialize the number of particles n, and use a random function to initialize the position and speed of each particle;
5: Calculate the F_1-Score of each particle;
6: Initialize the optimal parameter $\overline{\text{pbest}}$ of each particle, and the optimal parameter $\overline{\text{gbest}}$ of the population according to F_1-Score;
7: Initialize the iteration number $k = 1$;
8: Use (2) and (3) to update the position and speed of each particle;
9: Update the value of F_1-Score;
10: Update the historically optimal parameters of each particle;
11: Update the globally optimal parameters of the population;
12: $k = k + 1$
13: **if** $k > k_{\max}$ **then**
14: Output the optimal position vector and the corresponding F_1-Score value;
15: **else**
16: Go back to Step 8;
17: **end if**

4.2 Metric

This paper uses three metrics, i.e., precision, recall and F_1-Score, to evaluate the performance of our proposed scheme.

Here, according to the correct and wrong results of classifier's prediction on the test dataset, it can be divided into three situations:

- TP (True Positive): the entities in test dataset and they are recognized by model.
- FP (False Positive): the entities recognized by model while they are not existing in test dataset.
- FN (False Negative): the entities existing in datasets but they are not recognized by model.

Then, we can get the definition of three metrics as follows:

- Precision (P): the proportion of positive predictions that are correct in all predictions.
- Recall (R): the proportion of all positive samples that are correctly predicted to be positive.
- F_1-Score (F): Harmonized average of Precision and Recall.

Furthermore, they are shown as:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

$$F = \frac{2 \times P \times R}{P + R}. \tag{6}$$

4.3 Experimental Comparison

We use BERT-CRF [21], BERT-BiLSTM-CRF [35] and our BERT-CRF-PSO to conduct experiments on three datasets described in Sect. 4.1. By consulting the literature, we found that for fine-tuning, except for batch_size, training_epoch and learning_rate, most model hyperparameters are the same as in pre-training [12]. Where batch_size is the number of samples sent to the network for training each time, training_epoch refers to the process of sending all data to the network to complete a forward calculation and back propagation, and learning_rate controls the learning speed of the model. Therefore, in the experiment, we choose the above three hyperparameters as the objects to be automatically fine-tuned by the PSO algorithm. In addition, in order to prove the effectiveness of the proposed method, we designed a comparative experiment of fine-tuning two hyperparameters (batch_size, training_epoch) and fine-tuning three hyperparameters (batch_size, training_epoch and learning_rate). On the basis of experience and many experiments, w is set to 0.4, c_1 and c_2 are both set to 2. Meanwhile, batch_size $\in [8, 32]$, training_epoch $\in [10, 30]$ and learning_rate $\in$[1e−5, 1e−4] are three hyperparameters of BERT model, and they need to be adjusted with PSO algorithm.

Table 1 shows the comparison results of different methods on different datasets. Compared with other two models, our proposed scheme employs PSO to automatically search for the optimal value of hyperparameters in a given searching space, so that the effect of the model for a specific dataset reaches the global optimal. During the experiment, the hyperparameters of other two models are manually set by us in consideration of experience and the results of multiple experiments. The experimental results show that the model that requires manual adjustment of hyperparameters is not only time-consuming and labor-intensive, but also easy to fall into the local optimum. The proposed scheme can find the global optimal solution through collaboration and information sharing between individuals in the group. The results in Table 1 confirm the superiority of our scheme. Additionally, if we increase the dimensional number of position vectors, i.e., model hyperparameters in Algorithm 1, the improved performance of our scheme will be more obvious. The comparative experimental results of fine-tuning two hyperparameters and fine-tuning three hyperparameters in Table 1 prove this statement.

Specifically, in Table 2 we provide the optimal hyperparameters found by PSO in our scheme on the three datasets. Obviously, for different dataset, the optimal values of model hyperparameters are different. The experimental results show that the model we propose can help us automatically find the global optimal hyperparameters for a specific dataset, which is a very efficient method.

Table 1. Comparison results of different methods on different datasets.

Dataset	Method	Precision	Recall	F_1-Score
Boson	BERT-CRF	82.45%	85.14%	83.78%
	BERT-BiLSTM-CRF	82.69 %	85.42%	84.03%
	BERT-CRF-PSO(a)	**82.97%**	**85.98%**	**84.45%**
	BERT-CRF-PSO(b)	**83.78%**	**85.37%**	**84.57%**
People's daily	BERT-CRF	93.50%	95.63%	94.55%
	BERT-BiLSTM-CRF	92.29%	94.59%	93.43%
	BERT-CRF-PSO(a)	**93.27%**	**97.86%**	**95.51%**
	BERT-CRF-PSO(b)	**95.43%**	**98.29%**	**96.84%**
Inquiry data of the steel industry	BERT-CRF	91.39%	93.40%	92.39%
	BERT-BiLSTM-CRF	92.14%	94.57%	93.34%
	BERT-CRF-PSO(a)	**93.21%**	**94.57%**	**93.89%**
	BERT-CRF-PSO(b)	**92.84%**	**95.01%**	**93.91%**

Note: (a) represents fine-tuning two hyperparameters, (b) represents fine-tuning three hyperparameters.

Table 2. Comparison of optimal model hyperparameters of different datasets.

Dataset	batch_size	training_epoch	learning_rate
Boson	21	20	–
	12	24	3e−5
People's daily	24	22	–
	19	17	4e−5
Inquiry data of the steel industry	22	29	–
	24	24	2e−5

5 Conclusion

Through the combination of collaborative optimization mechanism, we design a novel scheme BERT-CRF-PSO to achieve entity extraction task in this paper. Then, three modules, including BERT, CRF and PSO algorithm, are incorporated into our scheme. Here, the BERT model is utilized to pre-train data, the CRF adds constraints to the predicted label to ensure its legitimacy, and the PSO fine-tunes the hyperparameters of the BERT module, which can automatically find the optimal value of the hyperparameters for a specific dataset in a cooperative way. These three modules cooperate with each other and share information to complete the entity extraction task. Hence, the training results within entire architecture can achieve the global optimum. The experimental results on two public datasets and a steel inquiry dataset confirm the effectiveness of our scheme, while improving the performance of entity extraction.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant U1836106, in part by the Beijing Natural Science Foundation under Grants 19L2029 and M21032, in part by the Scientific and

Technological Innovation Foundation of Shunde Graduate School, USTB, under Grants BK19BF006 and BK20BF010, and in part by the Fundamental Research Funds for the University of Science and Technology Beijing under Grant FRF-BD-19-012A.

References

1. Bashir, Z., El-Hawary, M.: Applying wavelets to short-term load forecasting using PSO-based neural networks. IEEE Trans. Power Syst. **24**(1), 20–27 (2009)
2. de Bruijn, B., Cherry, C., Kiritchenko, S., Martin, J., Zhu, X.: Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. J. Am. Med. Inform. Assoc. **18**(5), 557–562 (2011)
3. Cai, J., Wei, H., Yang, H., Zhao, X.: A novel clustering algorithm based on DPC and PSO. IEEE Access **8**, 88200–88214 (2020)
4. Cao, P., Chen, Y., Liu, K., Zhao, J., Liu, S.: Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 182–192. Association for Computational Linguistics, Brussels, Belgium (2018)
5. Carbonell, M., Riba, P., Villegas, M., Fornés, A., Lladós, J.: Named entity recognition and relation extraction with graph neural networks in semi structured documents. In: Proceedings of the 25th International Conference on Pattern Recognition, pp. 9622–9627. IEEE, Milan, Italy (2020)
6. Chen, M., Shen, H., Huang, Z., Luo, X., Yin, J.: Towards accurate search for e-commerce in steel industry: a knowledge-graph-based approach. In: Gao, H., Wang, X., Iqbal, M., Yin, Y., Yin, J., Gu, N. (eds.) CollaborateCom 2020. LNICST, vol. 349, pp. 3–18. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67537-0_1
7. Chiu, J., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Trans. Assoc. Comput. Linguist. **4**, 357–370 (2016)
8. Chopra, D., Joshi, N., Mathur, I.: Named entity recognition in Hindi using hidden Markov model. In: Proceedings of the Second International Conference on Computational Intelligence & Communication Technology, pp. 581–586. IEEE, Ghaziabad, India (2016)
9. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 100–110. Association for Computational Linguistics, MD, USA (1999)
10. Constant, M., Sigogne, A.: MWU-aware part-of-speech tagging with a CRF model and lexical resources. In: Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World, pp. 49–56. Association for Computational Linguistics, Oregon, USA (2011)
11. Das, A., Garain, U.: CRF-based named entity recognition @icon 2013. arXiv:1409.8008 (2014)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
13. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the Congress on Evolutionary Computation, vol. 1, pp. 81–86. IEEE, COEX, Seoul, Korea (2001)

14. Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C.D., Stamatopoulos, P.: Rule-based named entity recognition for greek financial texts. In: Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries, pp. 75–78. Citeseer (2000)
15. Khabsa, M., Giles, C.L.: Chemical entity extraction using CRF and an ensemble of extractors. J. Cheminformatics **7**(1), 1–9 (2015)
16. Khalifa, M.H., Ammar, M., Ouarda, W., Alimi, A.M.: Particle swarm optimization for deep learning of convolution neural network. In: Proceedings of the Sudan Conference on Computer Science and Information Technology, pp. 1–5. IEEE, Elnihood, Sudan (2017)
17. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning, pp. 282–289. Morgan Kaufmann, San Francisco, CA, USA (2001)
18. Lin, X., Peng, H., Liu, B.: Chinese named entity recognition using support vector machines. In: Proceedings of the International Conference on Machine Learning and Cybernetics, pp. 4216–4220. IEEE, Guangzhou, China (2006)
19. Lison, P., Barnes, J., Hubin, A., Touileb, S.: Named entity recognition without labelled data: a weak supervision approach. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1518–1533. Association for Computational Linguistics (2020)
20. Liu, L., et al.: Empower sequence labeling with task-aware neural language model. In: Proceedings of the 32th AAAI Conference on Artificial Intelligence, vol. 32. AAAI Press, Hilton New Orleans Riverside, New Orleans, Louisiana, USA (2018)
21. Liu, M., Tu, Z., Wang, Z., Xu, X.: LTP: a new active learning strategy for BERT-CRF based named entity recognition. arXiv:2001.02524 (2020)
22. Liu, Y., Zhang, Y., Che, W., Liu, T., Wu, F.: Domain adaptation for CRF-based Chinese word segmentation using free annotations. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 864–874. ACL, Doha, Qatar (2014)
23. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1064–1074. Association for Computational Linguistics, Berlin, Germany (2016)
24. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(24), 3–26 (2007)
25. Nadeau, D., Turney, P.D., Matwin, S.: Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. In: Lamontagne, L., Marchand, M. (eds.) AI 2006. LNCS (LNAI), vol. 4013, pp. 266–277. Springer, Heidelberg (2006). https://doi.org/10.1007/11766247_23
26. Omran, M.G.H., Engelbrecht, A.P., Salman, A.A.: Particle swarm optimization for pattern recognition and image processing. In: Abraham, A., Grosan, C., Ramos, V. (eds.) Swarm Intelligence in Data Mining. Studies in Computational Intelligence, vol. 34. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-34956-3_6
27. Qu, L., Ferraro, G., Zhou, L., Hou, W., Baldwin, T.: Named entity recognition for novel types by transfer learning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 899–905. The Association for Computational Linguistics, Texas, USA (2016)
28. Saha, S.K., Sarkar, S., Mitra, P.: Feature selection techniques for maximum entropy based biomedical named entity recognition. J. Biomed. Inform. **42**(5), 905–911 (2009)

29. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st Annual Conference on Neural Information Processing Systems, pp. 5999–6009. Neural information processing systems foundation, Long Beach, CA, USA (2017)
30. Wang, Q., Iwaihara, M.: Deep neural architectures for joint named entity recognition and disambiguation. In: Proceedings of the IEEE International Conference on Big Data and Smart Computing, pp. 1–4. IEEE, Kyoto, Japan (2019)
31. Wang, X., et al.: Cross-type biomedical named entity recognition with deep multi-task learning. Bioinformatics **35**(10), 1745–1752 (2019)
32. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, vol. 30, pp. 2659–2665. AAAI Press, Phoenix, Arizona USA (2016)
33. Zhang, D., et al.: Improving distantly-supervised named entity recognition for traditional Chinese medicine text via a novel back-labeling approach. IEEE Access **8**, 145413–145421 (2020)
34. Zhang, S., Elhadad, N.: Unsupervised biomedical named entity recognition: experiments with clinical and biological texts. J. Biomed. Inform. **46**(6), 1088–1098 (2013)
35. Zhang, W., Jiang, S., Zhao, S., Hou, K., Liu, Y., Zhang, L.: A BERT-BILSTM-CRF model for Chinese electronic medical records named entity recognition. In: Proceedings of the 12th International Conference on Intelligent Computation Technology and Automation, pp. 166–169. IEEE, Xiangtan, China (2019)
36. Zhao, Z., et al.: ML-CNN: a novel deep learning based disease named entity recognition architecture. In: Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, pp. 794–794. IEEE, Shenzhen, China (2016)
37. Zhou, H., Sun, G., Fu, S., Liu, J., Zhou, X., Zhou, J.: A big data mining approach of PSO-based BP neural network for financial risk management with IoT. IEEE Access **7**, 154035–154043 (2019)
38. Žukov-Gregorič, A., Bachrach, Y., Coope, S.: Named entity recognition with parallel recurrent neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 69–74. Association for Computational Linguistics, Melbourne, Australia (2018)