Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2021.017491

ARTICLE

A Novel Named Entity Recognition Scheme for Steel E-Commerce Platforms Using a Lite BERT

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

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China

[2]Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, 100083, China

[3]Shunde Graduate School, University of Science and Technology Beijing, Foshan, 528399, China

[4]Ouyeel Co., Ltd., Shanghai, 201999, China

[*]Corresponding Author: Xiong Luo. Email: xluo@ustb.edu.cn

Received: 13 May 2021 Accepted: 12 June 2021

ABSTRACT

In the era of big data, E-commerce plays an increasingly important role, and steel E-commerce certainly occupies a positive position. However, it is very difficult to choose satisfactory steel raw materials from diverse steel commodities online on steel E-commerce platforms in the purchase of staffs. In order to improve the efficiency of purchasers searching for commodities on the steel E-commerce platforms, we propose a novel deep learning-based loss function for named entity recognition (NER). Considering the impacts of small sample and imbalanced data, in our NER scheme, the focal loss, the label smoothing, and the cross entropy are incorporated into a lite bidirectional encoder representations from transformers (BERT) model to avoid the over-fitting. Moreover, through the analysis of different classic annotation techniques used to tag data, an ideal one is chosen for the training model in our proposed scheme. Experiments are conducted on Chinese steel E-commerce datasets. The experimental results show that the training time of a lite BERT (ALBERT)-based method is much shorter than that of BERT-based models, while achieving the similar computational performance in terms of metrics precision, recall, and F_1 with BERT-based models. Meanwhile, our proposed approach performs much better than that of combining Word2Vec, bidirectional long short-term memory (Bi-LSTM), and conditional random field (CRF) models, in consideration of training time and F_1.

KEYWORDS

Named entity recognition; bidirectional encoder representations from transformers; steel E-commerce platform; annotation technique

1 Introduction

The past two decades have witnessed the rapid advancements of E-commerce technologies, and steel E-commerce plays an important role in this field. With the increasingly large-scale in online procurement of steel raw materials, the corresponding E-commerce platforms have accumulated massive data. If these online transaction data can be analyzed and mined to extract

useful information with natural language processing (NLP) techniques, it can not only improve the efficiency of customers purchasing steel online, but also promote the service of E-commerce platforms. However, considering the diversity of steel categories, it is a huge challenge to provide satisfactory search results to steel buyers on steel E-commerce platforms. Hence, how to utilize those enormous data to mine and analyze information, so as to improve the procurement efficiency of consumers is an urgent problem to be dealt with. With the steel E-commerce platform in China as an example, Chinese steel turnover exceeded 100 million tons in 2019 [1]. However, because of the characteristics of Chinese text and the strong professionalism in the steel industry, it took much time and effort for professionals to extract key entities to analyze buyers' requirements manually. Thus, it is necessary to extract important information from buyers' demands intelligently [2].

Named entity recognition (NER), known as entity extraction, is a primary intelligent technology in NLP. It focuses on extracting information units, such as names (including person, organization and location names), and numeric expressions (including date, money, time and percent expressions) from unstructured text [3]. Early, language experts constructed rules or templates manually to form dictionary or rule-based methods [4], which choose key words, direction words, place words and some others as characteristics. Then, pattern and string matching is main method to extract entities [5]. Some problems of those methods are that they rely heavily on the establishment of knowledge rules and dictionaries. In addition, it is very timing-consuming for formulating rules and dictionaries, and it needs experts to reconstruct rules and dictionaries for different fields [6]. Traditional machine learning methods for NER are based on statistical learning, such as hidden Markov model (HMM) [7], support vector machine (SVM) [8], conditional random field (CRF) [9]. These methods regard NER as a sequence labeling problem and vast corpus are used to train a model to label each position in the sentence. Similarly, the main limitation of these methods is the requirement of building feature engineering, which is a timing-consuming task.

In recent years, deep learning has achieved great success in the field of NLP, such as text classification [10] and machine translation [11]. Different from traditional machine learning models, deep learning-based methods can extract text features automatically without too much human intervention [12]. Experiments have verified that mature deep learning methods can improve the performance of E-commerce platforms in other fields [13], however, there are still many challenges when applying them to extract entities in the steel E-commerce platforms. Specifically, the text data of steel E-commerce field are unstructured and contain complex technical terms and professional nicknames [14], and it is difficult to train a satisfactory model with few-shot data and imbalanced corpus.

Hence, it is necessary to exploit other advanced deep learning models. In 2017, transformer was proposed for capturing semantic features and it performs well than recurrent neural network (RNN) and convolutional neural network (CNN) [15]. Motivated by this model, many transformer-based pre-training models have been applied to represent text. Generative pre-trained transformer (GPT) uses transformer decoders to achieve better performance than previous models [16], but it only considers the influence of the leftward context and cannot learn bidirectional information. Bidirectional encoder representations from transformers (BERT) uses transformer encoders to capture bidirectional contextual representations [17], therefore it outperforms GPT. With the help of BERT model, many improved models have been developed, e.g., enhanced representation through knowledge integration (ERNIE) from Baidu [18], SpanBERT [19], robustly optimized BERT pre-training approach (RoBERTa) [20]. However, these pre-training models will generate massive parameters during training, which will take a lot of time to train good models.

To overcome this limitation, a lite BERT (ALBERT) model is developed to reduce the number of parameters extremely to speed up the training time, and it does not degrade the performance of model [21]. Considering the computational efforts, ALBERT is applied to extract entities in the steel E-commerce platforms in this article. Meanwhile, in order to address the issue of imbalanced and small data, and avoid the limitation of over-fitting, we integrate focal loss [22] and label smoothing [23] to cross entropy loss function as a new loss function to improve the performance of model. Moreover, some classic annotation schemes are analyzed for choosing a suitable one to tag the steel E-commerce data in our proposed method.

The contributions of this article are as follows:

(1) Aiming at the practical demand from buyers in the steel E-commerce platforms, an efficient deep learning-based intelligent model is employed to recognize technical terms.
(2) Considering the impacts of few-shot and imbalanced data, focal loss, label smoothing, and cross entropy are combined as a novel loss function to improve the performance of the ALBERT-based model.
(3) To achieve a good generalization performance, several annotation schemes are explored to analyze their influences on the NER, and an ideal annotation scheme is chosen to tag data in the Chinese steel E-commerce platforms.

The rest of this article is organized as follows. Section 2 introduces some related works about deep learning models and popular annotation schemes in NER task. In Section 3, the proposed scheme is presented in detail. Additionally, the experimental results and discussion on the Chinese steel E-commerce dataset are provided in Section 4. Finally, the conclusion and future work are summarized in Section 5.

2 Related Work

NER is a basic and essential task in NLP [24], which is widely used in text mining, machine translation, recommendation system, knowledge graph, and other domains [25]. How to design a NER method to achieve excellent performance is an important work in NLP. In this section, we introduce some typical technologies in relation to NER with imbalanced and few-shot corpus.

Recently, deep learning algorithms are usually developed to represent text and recognize entities. Mikolov et al. [26] proposed a Word2Vec model for computing word vectors. This model converts a word into a low-dimensional vector by using the contextual information of the word. The more resemble words are, the closer they are in vector space. Word vector is a remarkable milestone in NLP, and it has been widely used in Chinese word segmentation, sentiment classification, dependency parsing [27]. However, owing to one-to-one correspondence between words and vectors, it is difficult for Word2Vec model to represent polysemous words [28]. RNN was applied in NER, but with the existence of vanishing gradient and exploding gradient problems, this method was difficult to process long sequence text [29]. Long short-term memory (LSTM) as a variation of RNN [30], overcomes the limitation mentioned above, but it takes a lot of time to train the model several times for better convergence performance. Gate recurrent unit (GRU) has one less gate cell than LSTM [31], which is used to reduce the complexity of the model while its performance is similar to LSTM. As an optimized model of LSTM, lattice LSTM added a new word cell to represent the recurrent state from the beginning of a sentence [32]. The introduction of this structure effectively improves NER performance, but low-performance computing and poor portability are its major disadvantages.

For developing a satisfying model for NER, some different algorithms are combined to extract entities together. A classic architecture was proposed by Lample et al. [33], its input layers are vector representations of individual words using skip-n-gram. Then, the character embeddings of words are given to a bidirectional LSTMs (BiLSTM). Considering neighboring tags, this architecture adopted a CRF algorithm instead of the softmax out from BiLSTM, achieving the final predictions for every word. This method is a sentence-level NER method, suffering from the tagging inconsistency problem. To deal with the problem, Luo et al. [34] added an attention mechanism to BiLSTM-CRF for automatically ensuring tagging consistency in a document. Moreover, there are plenty of integrated architectures in the field of NER, such as CNN merged with LSTM-CRF [35], language model combined with LSTM-CRF model [36].

Pre-training language representations, such as embeddings from language models (ELMo) [37] and OpenAI GPT, have been shown to be effective for improving many NLP tasks. During these years, fine-tuning a pre-training language model has become an advanced method for learning semantic information in NLP [38]. Inspired by OpenAI GPT and ELMo, the state-of-the-art language model pre-training technique is BERT proposed by Devlin et al. [17]. As a strong text representation model, BERT can more capture long semantic information in a sentence. Many studies have demonstrated that BERT works better than previous models in NER. Mostly, increasing model size when pre-training language model often improves performance on downstream tasks. Considering computing memory limitations and longer training times, it becomes harder to further increase model size. To reduce memory consumption and speed-up the training speed of BERT, Lan et al. [21] presented ALBERT model whose scale is much smaller compared to the original BERT. Since ALBERT has learned good semantic relationships of text on a massive corpus, and its scale is enough smaller than other pre-training models, it is employed to extract entities from steel E-commerce data in this article.

Currently, available deep learning models are performing better on balanced dataset [39,40]. However, the stability and generalization of models may degrade when the training data are imbalanced [41]. In NER, data imbalance shows that a great difference in the number of entities in different categories. Generally, models perform well on labels with sufficient samples and perform awfully poorly on labels with fewer samples. There are various techniques and methods developed to deal with the problem of data imbalance. Traditional approaches are sampling methods, including over-sampling and under-sampling. Synthetic minority over-sampling technique (SMOTE) [42] and adaptive synthetic (ADASYN) [43] as typical over-sampling methods generate new samples added to small labels for balancing data distribution. But these methods may introduce extra noises into data and change the original data distribution. Focal loss was presented by Lin et al. [22], which assigned high weight for small sample labels to pay more attention to difficult-to-predict samples. Szegedy et al. [23] proposed label smoothing to achieve a good generalization performance, which preventing the model assign a full probability to each training example for avoiding the limitation of over-fitting.

In addition, a proper annotation scheme of part-of-speech (POS) is an essential component of NER. Tkachenko et al. [44] used BIO and BILOU tagging schemes for NER on Estonian, which showed that BILOU performed slightly better than BIO. However, Konkol et al. [45] showed that BILOU performed the worst in NER on four languages (English, Dutch, Czech and Spanish) than BIO-1, BIO-2, IEO, and BIEO annotation schemes. Malik et al. [46] proposed BIL2 annotation scheme, which was similar to BIESO, and it achieved the highest F-measure against IO, BIO-2, and BILOU. Mozharova et al. [47] applied IO and BIO tagging schemes on Russian text, and BIO outperformed IO tagging scheme. Alshammari et al. [48] indicated that the IO annotation scheme

surpassed other schemes, such as BIO, BIES, on Arabic. Due to different languages with their own structures, appropriate tagging schemes should be selected according to the characteristics of languages to achieve the best performance for NER task.

Motivated by previous works, this article designs a novel scheme to address the issues of ALBERT-based models on imbalanced and few-shot corpus for NER in the steel E-commerce field. In order to improve the performance of training model, some different annotation schemes are analyzed and the most satisfactory annotation scheme is used to tag the steel E-commerce data.

3 The Proposed Scheme

In this section, we will introduce the proposed scheme. It is able to achieve a better performance in the NER task with imbalanced and few samples. The structure of the ALBERT-based model is first presented. Then, the details of the loss function combined with focal loss, label smoothing, and cross entropy applied to avoid the limitation of over-fitting is described. Lastly, different annotation schemes are analyzed to improve the performance and an ideal tagging scheme is utilized to tag data in the end.

3.1 NER Using ALBERT-Based Model

Current state-of-the-art pre-training language models usually have hundreds of millions or more of parameters, and limited computing memory makes it difficult to experiment with large models. Then, BERT model uses Transformer encoders which can better learn deep bidirectional semantic information from large text, considering both left and right context in all layers. The architecture of the standard BERT model is shown in Fig. 1.

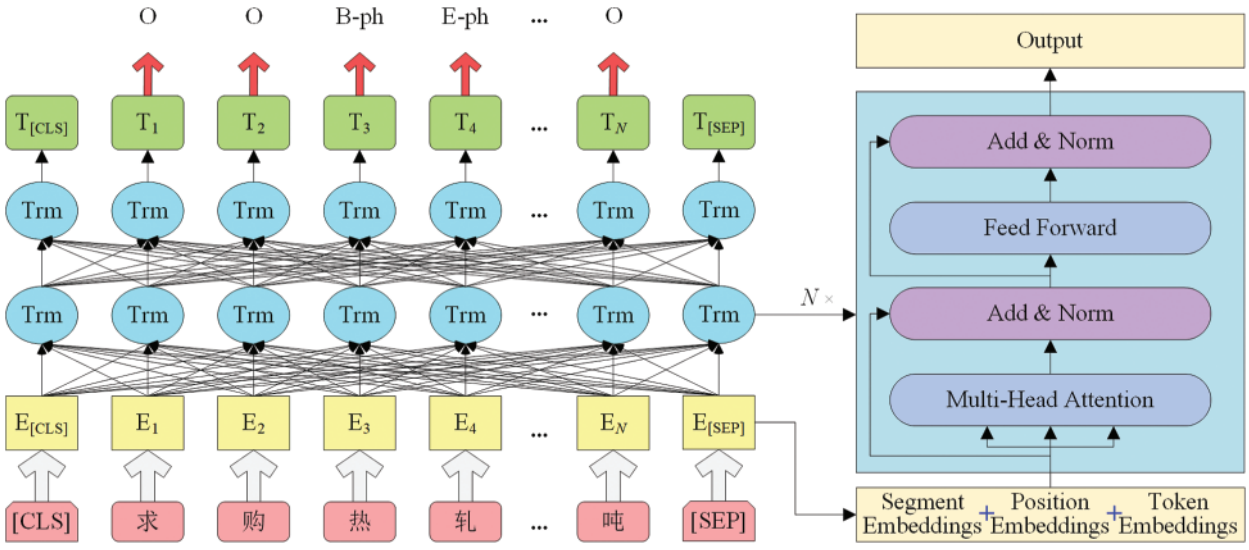

Figure 1: The architecture of a standard BERT model

As shown in this figure, the Chinese in the first layer are input sentences. In the second layer, BERT introduces segment and position embeddings of each word, and regards the summing of them and corresponding token embeddings as the input embeddings of each word, where N is the length of the input sentence. In NER, the "[CLS]" token represents the beginning of a sentence,

and the "[SEP]" token means the end of that sentence. All of "Segmentation Embeddings" are set to "0" because there only need one sentence in NER. "Position Embeddings" show a position feature of each word in a sentence. Moreover, "Token Embeddings" are obtained by one-hot word embeddings. "Trm" is an encoder structure with multi-head self-attention mechanism, position-wise feed-forward networks, layer normalization and residual connection, dispensing with recurrence and convolutions [15]. The outputs of BERT model are tags of each word in the input sentence. For example, "O" indicates that corresponding word which does not belong to any entity, "B-ph" means that the word is the beginning of entity and the category of this entity is "ph" which represents "Grade" in our dataset. Similarly, "E-ph" shows that the position of this word in the entity is the end and its entity class is "Grade" too. This architecture can learn more features of the text better. Generally, there are two steps for training BERT model, i.e., pre-training and fine-tuning. During pre-training, the model is trained with large unlabeled data. Then, all of the parameters of BERT model are fine-tuned with labeled steel E-commerce corpus from NER task. In this process, an enormous number of parameters will be generated.

A model with a large number of parameters will occupy a lot of memory and decrease the training speed. For reducing the number of parameters, ALBERT model was presented using two parameter-reduction techniques, i.e., factorized embedding parameterization and cross-layer parameter sharing. Following the BERT notation conventions, this article denotes the size of vocabulary as V, the size of input embeddings as E, and the size of hidden as H. In BERT, the embedding size E is always the same as the hidden size H. In order to include as many characters as possible, V is very large in NLP. Meanwhile, E will increase as increasing H, making the embedding matrix increasing too large, which is with size $E \times V$.

For reducing parameters size, ALBERT model unties E from H where $H \gg E$, and decomposes embedding matrix into two small matrices with factorization method. The specific method is projecting one-hot word embeddings into a low-dimensional vector space with size E, and then projecting them to the hidden space with size H rather than projecting them into the hidden space directly. The difference between BERT and ALBERT in the process of generating token embeddings is shown in Fig. 2. It is obvious that through this method, the number of embedding parameters can be reduced from $O\ (V \times H)$ to $O\ (V \times E + E \times H)$, where $E \ll H$.

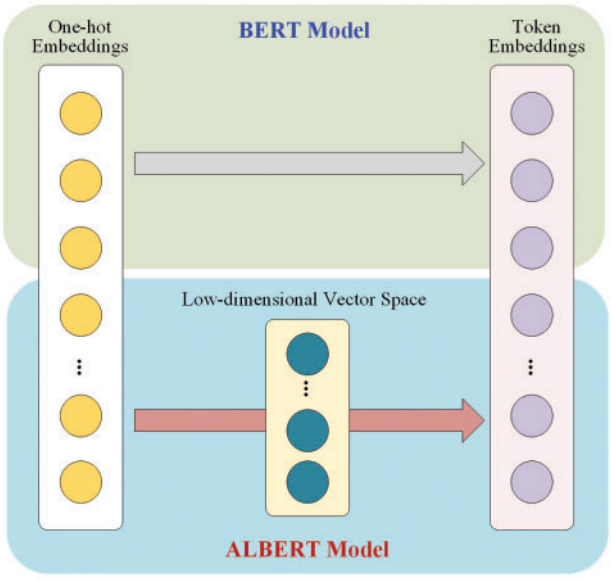

Figure 2: The difference between BERT and ALBERT in the process of generating token embeddings

Additionally, ALBERT shares all parameters across layers to improve the efficiency of model training. Meanwhile, assuming the number of encoder layers is L in the canonical BERT model, the number of parameters for all encoder layers can be reduced to $\frac{1}{L}$ through the use of ALBERT. Actually, the experiments evaluated on the cosine similarity and L_2 distances show that parameter sharing has a positive influence on stabilizing networks.

According to those two parameter-reduction techniques, the comparisons of the number of parameters between BERT and ALBERT models at different scales are shown in Tab. 1. From this table, it is clear that when BERT and ALBERT have the same values of layer number and hidden size, and just the embeddings sizes are different, the number of parameters of ALBERT model is far less than that of BERT model. It is noted that the ALBERT model only reduces the number of parameters to decrease the space complexity, and does not change the calculation time. Thus, its time complexity is the same as that of the BERT model.

Table 1: The comparison of the number of parameters between BERT and ALBERT models at different scales

Model	Layers number	Hidden number	Embeddings number	Heads number	Parameters number
BERT-base	12	768	768	12	110 M
BERT-large	24	1024	1024	16	340 M
ALBERT-base	12	768	128	12	12 M
ALBERT-large	24	1024	128	16	18 M
ALBERT-XLarge	24	2048	128	32	60 M
ALBERT-XXLarge	12	4096	128	64	235 M

3.2 Avoiding the Over-Fitting by a Novel Loss Function

The last layer outputs of ALBERT is the score between each word and each label. Then, the probability can be obtained through the softmax function. Finally, the predicted probability distribution $\hat{y}$ of model and the one-hot probability distribution of the ground-truth labels y are calculated with a standard cross entropy as loss function shown in (1):

$$CE\left(y,\hat{y}\right) = -\sum_{\forall x} y\left(x\right)\log\left(\hat{y}\left(x\right)\right) = -\frac{1}{N}\left(\sum_{i=1}^{N} y_i \cdot \log\left(\hat{y}_i\right)\right), \tag{1}$$

where N is the length of a sentence, x represents each word in the sentence.

However, this method just interests in the accuracy of prediction probability of correct labels, and ignores the prediction results of other labels, which leads to a scatter of learned features. Meanwhile, the data imbalance will affect the correlation between data features. In this case, the trained model with cross entropy will have a bias resulting to perform well on balanced corpus and perform badly on imbalanced data.

To deal with the above problems, Lin et al. [22] proposed the focal loss, which reshaped the standard cross entropy by increasing the loss weight assigned for the labels with fewer samples, thus the model focused on training hard-predicted samples. Adding weighting factor α and modulating factor $\left(1-\hat{y}_i\right)^{\gamma}$ with tunable focusing parameter γ, the focal loss is defined as (2):

$$FL\left(y,\hat{y}\right)=-y\left(x\right)\alpha\left(1-\hat{y}\left(x\right)\right)^{\gamma}\log\left(\hat{y}\left(x\right)\right), \tag{2}$$

where $\gamma\in[0,\ 5]$ and $\alpha\in[0,\ 1]$ are two hyperparameters. Here, when $\gamma=0$ and $\alpha=1$, the focal loss becomes cross entropy.

This method decreases the loss weight of easy-to-predict samples, and emphasizes the weight on the hard or mispredicted samples. Although it can achieve excellent results on NER by focal loss-based ALBERT model, there is a problem of small sample size in steel E-commerce platform. It is necessary to construct corpus manually with the help of experts, but there will be some noises generated in the training data due to carelessness. During constructing corpus, one-hot embeddings are used to represent ground-truth labels. During training process, if the label value is "1", the corresponding prediction value can be remained to calculate loss, and all other predicted values will be discarded. This mechanism makes the model awfully confident of the tagged labels, which leads to encourage a growing distance between the maximum predicted value and other predicted values. To address this issue, this article combines label smoothing, focal loss, and cross entropy as a new loss function for ALBERT model.

Label smoothing was first proposed by Szegedy et al. [23], which aims at encouraging the model to be less confident during training. Label smoothing introduced a smoothing parameter $\varepsilon\in(0,\ 1)$, and it is shown as (3):

$$LS\left(y\right)=\left(1-\varepsilon\right)y\left(x\right)+\frac{\varepsilon}{K}, \tag{3}$$

where $y(x)$ is the one-hot embeddings of ground-truth labels, and K is the number of labels. The original truth label will be decreased from "1" to "1 $-\varepsilon$", and the other label values will be replaced "0" with "$\frac{\varepsilon}{K}$".

According to (1)–(3), the whole loss function in the model can be defined as (4):

$$\mathfrak{L}\left(y,\hat{y}\right)=-\sum_{\forall(x)}\left(\left(1-\varepsilon\right)y\left(x\right)+\frac{\varepsilon}{K}\right)\alpha\left(1-\hat{y}\left(x\right)\right)^{\gamma}\log\left(\hat{y}\left(x\right)\right)-\sum_{\forall x}y\left(x\right)\log\left(\hat{y}\left(x\right)\right). \tag{4}$$

In our scheme, the loss function will replace the cross entropy out from the standard ALBERT model, and the whole architecture of our proposed scheme is shown in Fig. 3.

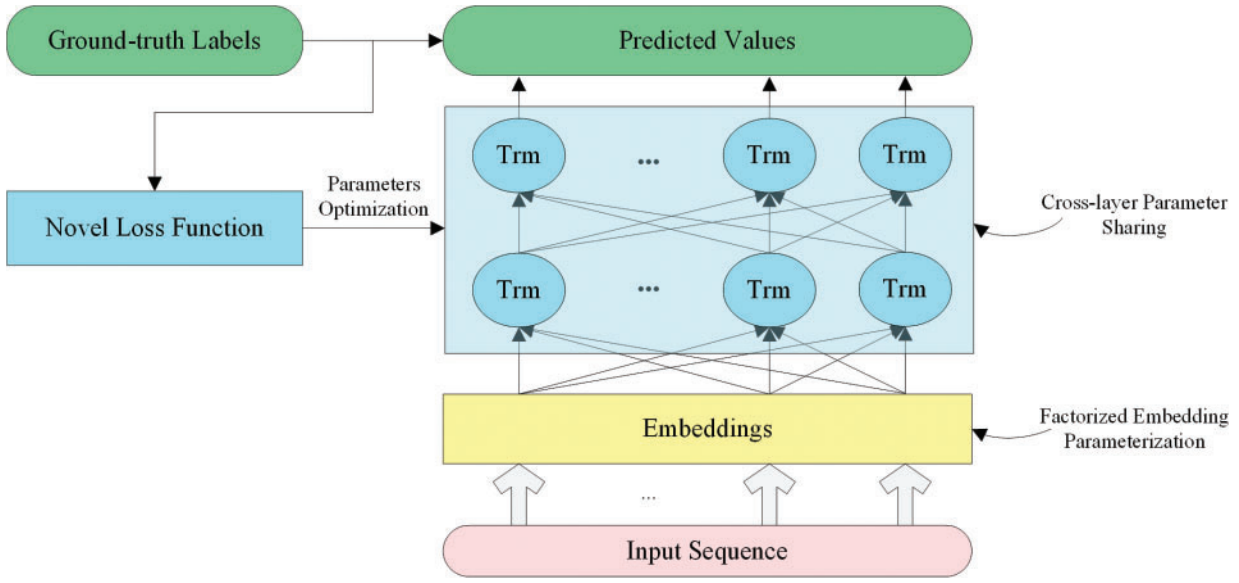

Figure 3: The architecture of the proposed scheme

3.3 The POS with Different Annotation Schemes

How to choose an appropriate annotation scheme is a meaningful task for NER. Currently, many popular annotation schemes on NER have been used in different works and those schemes are as follows:

(1) IO. This scheme is a relatively simple tagging scheme in NER task. There are only two tags in this scheme, including an inside tag (I) and an outside tag (O). For each word in entities, there are tagged with "I", and others tagged with "O".
(2) BIO. This scheme is a little complex than IO tagging scheme. Three tags are assigned for text to estimate whether the word is the beginning (B) of a named entity, inside (I) of a named entity, or outside (O) of all named entities. Additionally, BI tagging scheme is similar to BIO annotation scheme.
(3) EIO. This scheme works almost identically to BIO annotation, but it represents the end of the entity (E) rather than its beginning. Similarly, the EI tagging scheme is alike to the EIO annotation scheme.
(4) BIES. On the basis of BIO and EIO tagging schemes, BIES annotation scheme was proposed to further tag words, which leads to a great increase in the number of tags. In this scheme, "B" represents that the word is at the beginning of position of the entity, "I" indicates that the word is at the inside position of the entity, "E" shows that the word is at the end of the entity, and "S" means that the single word can be regarded as an entity.
(5) BIESO. As an alternative to the BIES annotation scheme, this tagging scheme integrals BIO and BIES schemes and it has the largest amount of tags than others. BIESO uses "B" to tag the beginning of a named entity, "I" to tag the inside of a named entity, "E" to tag the end of a named entity, "S" for entities with a single word, and "O" to tag others.

The number of tags generated by different annotation schemes are different. Assuming the number of entity categories (such as person name, location name, and time) is C in datasets,

the corresponding tags number are shown in Tab. 2. From this table, we can see that the BIESO annotation scheme generates the largest number of tags than others.

Table 2: The number of tags generated by different annotation schemes

Annotation scheme	The number of tags
IO	$C + 1$
BIO	$2C + 1$
EIO	$2C + 1$
BIES	$3C + 1$
BIESO	$4C + 1$

4 Experiment Results and Discussion

Different from public datasets in NER, the data size in steel E-commerce platform are smaller, and it is more professional and complex than other common datasets. In this section, we will detail the experimental results with the Chinese steel E-commerce text data. Especially, to evaluate the performance of our proposed scheme, we compare it with some other popular models, such as the BERT-based models and standard ALBERT model. In addition, our experiments are performed in the Python 3.7.9 environment running on the computer with the Ubuntu 19.10 system.

4.1 Datasets and Experiment Description

There are more than 3000 customers' history purchase information in original datasets. Through data cleaning and deduplication, 1322 useful data are remained to use. The datasets used in our experiments are actual and reasonable, and they are labeled using different annotation schemes with the help of steel experts. Considering that there is no entity composed of single word in text data and EIO works nearly identically to BIO, hence, we use IO, BIO and BIEO annotation schemes to tag data.

The key entity class in the steel E-commerce platform can be defined into 7 categories, and the data label distributions in our experiments are shown in Fig. 4. From this figure, we can see that the data scale of the "Surface Treatment" category is much smaller than that of the "Grade" category, resulting in an imbalanced distribution of data.

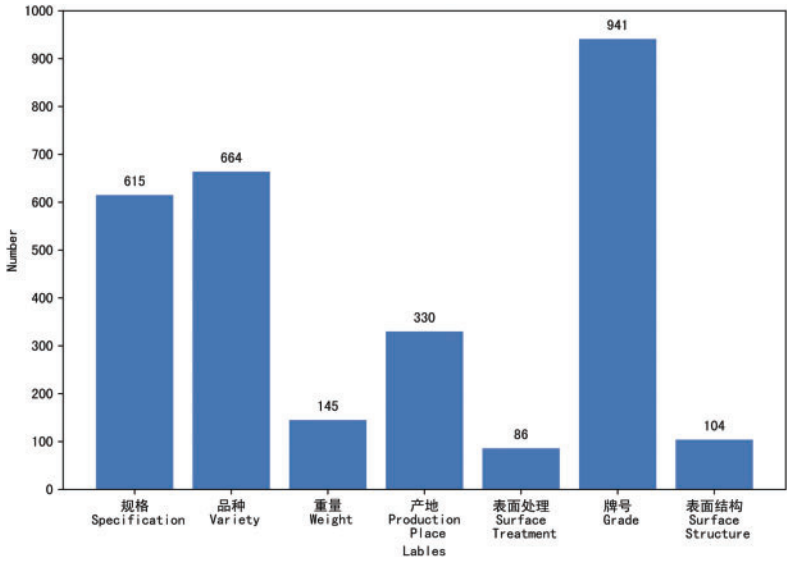

Figure 4: The data label distributions in datasets

4.2 Metrics

To evaluate the performance of different NER models with various annotation schemes, three metrics are employed. Here, 'Precision' represents the ratio of the number of correct entities recognized by model to the number of all entities recognized by the model, 'Recall' means the ratio of the number of correct entities recognized by model to the number of all entities that the model should recognize, and F_1 is a metric that merges the 'Precision' and 'Recall' of the model.

Those three metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \times 100\%, \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \times 100\%, \tag{6}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{7}$$

where TP is the number of correct entities recognized by model, FP is the number of wrong entities recognized by model, and FN is the number of entities in the data that should be recognized by the model but not recognized.

4.3 Performance Comparison

The Chinese steel datasets are tagged with the help of experts, and then, they are further tagged using different annotation schemes. Then, the novel loss function is employed in the pre-trained ALBERT model. In these experiments, on the basis of the pre-training parameters of ALBERT model, max sequence length is set to 128, batch size is set to 8, learning rate is set to 2×10^{-5}, and train epoch is set to 30. Motivated by [22,23], the hyperparameters α, γ, and ε are set to 0.25, 2, and 0.1, respectively.

Moreover, the datasets are trained using ALBERT-based model with different loss functions. The test results with different annotation schemes and loss functions are shown in Tab. 3. As can be seen from this table, the method we proposed can always achieve the best performance using each annotation scheme compared to the other loss functions. Among the three annotation schemes, the IO tagging scheme is the most proper for the Chinese steel E-commerce data than the BIO and BIEO tagging schemes. In addition, the loss function combined with label smoothing, focal loss, and cross entropy performs better than cross entropy, focal loss, label smoothing-cross entropy.

With a novel loss function, i.e., label smoothing-focal loss-cross entropy, the results of NER for each category using ALBERT model with IO annotation scheme are shown in Tab. 4. From this table, we can see that although the data size of "Surface Treatment" is much smaller than that of others, the result for its F_1 is still acceptable. The reason of this result is that the proposed scheme pays more attention to few-shot "Surface Treatment" samples. Hence, the combination of focal loss, label smoothing and cross entropy can effectively cope with the issue of data imbalance in small Chinese steel E-commerce data.

Table 3: The test results of various loss functions using different annotation schemes on Chinese steel E-commerce dataset in use of ALBERT-based model

Tagging scheme	Metric	Cross entropy	Focal loss	Label smoothing-cross entropy	Label smoothing-focal loss-cross entropy
IO	Precision	85.37%	84.27%	87.70%	**90.00%**
	Recall	88.77%	87.32%	89.13%	**91.30%**
	F_1	0.87	0.86	0.88	**0.91**
BIO	Precision	83.91%	83.84%	84.76%	**88.16%**
	Recall	88.15%	85.89%	88.15%	**89.55%**
	F_1	0.86	0.85	0.86	**0.89**
BIEO	Precision	83.33%	80.00%	82.92%	**86.77%**
	Recall	88.85%	87.80%	87.11%	**90.24%**
	F_1	0.86	0.84	0.85	**0.88**

Table 4: The test results of NER for each category using ALBERT model with IO annotation scheme on Chinese steel E-commerce dataset

Entity category	Precision (%)	Recall (%)	F_1
Surface treatment	100.00	71.43	0.83
Surface structure	94.12	100.00	0.97
Production place	87.88	89.23	0.89
Specification	94.12	95.73	0.95
Grade	89.29	90.91	0.90
Variety	86.62	89.78	0.88
Weight	92.11	92.11	0.92

Furthermore, to further evaluate the performance of our scheme, more experiments are conducted on another dataset with the same features. Here, we select the dataset named 'The PFR People's Daily corpus',[1] whose scale is the same as that of the Chinese steel E-commerce dataset, and there are 1358 place names, 1221 people names and 145 organization names. The test results with different annotation schemes and loss functions are shown in Tab. 5. It can be seen from this table that our proposed scheme can achieve better performance than others on 'The PFR People's Daily corpus'.

Finally, to further verify the performance of the proposed scheme, we compare it with some other popular models, including the classic Word2Vec-BiLSTM-CRF model, the BERT-CRF model, and BERT-BiLSTM-CRF model. Those models all use the same datasets tagged with IO annotation scheme. The result is shown in Tab. 6, where all training epochs are set to 30. From this table, we can see that the training time of our proposed scheme is about 250 s, and that of the BERT-based model is more than 720 s. Obviously, the training time of our proposed ALBERT-based scheme is much shorter than that of BERT-based models, while achieving the similar computational performance in terms of Precision, Recall and F_1. Since the embeddings

[1] http://www.ling.lancs.ac.uk/corplang/pdcorpus/pdcorpus.html

size in our model is set to 128, while it is set to 768 in the BERT-based model. In this step, the number of parameters can be reduced from 16226304 (21128×768) to 2802688 ($21128 \times 128 + 128 \times 768$), where 21128 is the number of vocabulary. Meanwhile, our model shares all parameters across 12 hidden layers, so that the number of parameters in this part can be reduced to $\frac{1}{12}$ of the BERT-based model. Through these two methods, ALBERT-based model can reduce massive parameters to decrease the space complexity and speed up training time, while it does not decrease computing capability. Meanwhile, compared with the classic model Word2Vec-BiLSTM-CRF, our scheme is more accurate than it and achieves better performance in training time, because the semantic feature extraction capability of bidirectional transformer significantly exceeds that of BiLSTM.

Table 5: The test results of various loss functions using different annotation schemes on 'The PFR people's daily cor*pus*' dataset in the use of ALBERT-based model

Tagging scheme	Metric	Cross entropy	Focal loss	Label smoothing-cross entropy	Label smoothing-focal loss-cross entropy
IO	Precision	88.07%	86.79%	85.33%	**92.40%**
	Recall	91.65%	90.93%	90.21%	**93.96%**
	F_1	0.90	0.89	0.88	**0.93**
BIO	Precision	86.07%	84.72%	87.19%	**89.40%**
	Recall	91.41%	89.98%	90.93%	**92.60%**
	F_1	0.89	0.87	0.89	**0.91**
BIEO	Precision	86.10%	84.11%	86.62%	**87.81%**
	Recall	91.65%	90.93%	91.17%	**92.84%**
	F_1	0.89	0.87	0.89	**0.90**

Table 6: The performance comparison between other models and ours using IO annotation scheme on Chinese steel E-commerce dataset

Metric	Word2Vec-BiLSTM-CRF	BERT-CRF	BERT-BiLSTM-CRF	Ours
Precision	27.26%	**90.88**%	89.54%	90.00%
Recall	21.01%	**93.84**%	91.49%	91.30%
F_1	0.24	**0.92**	0.91	0.91
Times (s)	418.26	720.39	726.71	**250.50**

Clearly, focal loss, label smoothing and cross entropy combined as a loss function in the ALBERT model can achieve a satisfactory performance of NER in the steel E-commerce platform, and IO annotation scheme is more suitable for the steel E-commerce data than BIO and BIEO tagging schemes. The training time of our model is much shorter than that of BERT-based models, while it achieves a similar computational performance in terms of Precision, Recall, F_1. Hence, considering the accuracy and the consuming time simultaneously, our proposed scheme may be a competitive choice in NER task for the steel E-commerce platform.

5 Conclusion

A novel loss function combined with focal loss, label smoothing and cross entropy, has been employed in ALBERT-based model to extract entities from the steel E-commerce data. In this proposed method, focal loss is used to address the issue of data imbalance, and label smoothing is utilized to encourage the model to be less confident to avoid over-fitting. Meanwhile, to achieve a good generalization performance of model, different annotation schemes are analyzed and an ideal tagging scheme is used to tag data. Experimental results show that our proposed scheme can efficiently deal with the issues of small sample and data imbalance, and it improves the performance for NER in the steel E-commerce platform. Thus, the proposed scheme is superior to BERT-based models in training time, and performs better than the Word2Vec-Bi-LSTM-CRF model in accuracy and time-consuming. Furthermore, IO annotation scheme is more suitable for the steel E-commerce data than BIO and BIEO annotation schemes.

In the future, to achieve a better performance, we will integrate some optimization algorithms into our proposed method to adjust some key hyperparameters. Moreover, we will employ our scheme in more fields and improve performance for NER of other fields.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grants U1836106 and 81961138010, in part by the Beijing Natural Science Foundation under Grants M21032 and 19L2029, in part by the Beijing Intelligent Logistics System Collaborative Innovation Center under Grant BILSCIC-2019KF-08, in part by the Scientific and Technological Innovation Foundation of Shunde Graduate School, USTB, under Grants BK20BF010 and BK19BF006, and in part by the Fundamental Research Funds for the University of Science and Technology Beijing under Grant FRF-BD-19-012A.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Conejo, A. N., Birat, J. P., Dutta, A. (2020). A review of the current environmental challenges of the steel industry and its value chain. *Journal of Environmental Management, 259(1),* 109782. DOI 10.1016/j.jenvman.2019.109782.
2. Gao, H., Qin, X., Barroso, R. J. D., Hussain, W., Xu, Y. et al. (2020). Collaborative learning-based industrial IoT API recommendation for software-defined devices: The implicit knowledge discovery perspective. *IEEE Transactions on Emerging Topics in Computational Intelligence.* 1–11. DOI 10.1109/TETCI.2020.3023155.
3. Sang, E. F., de Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *Proceedings of the Seventh Conference on Natural Language Learning,* pp. 142–147. USA: Association for Computational Linguistics.
4. Petasis, G., Vichot, F., Wolinski, F., Paliouras, G., Karkaletsis, V. et al. (2001). Using machine learning to maintain rule-based named-entity recognition and classification systems. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics,* pp. 426–433. Toulouse, France: Morgan Kaufmann.
5. Chen, H. H., Ding, Y. W., Tsai, S. C., Bian, G. W. (1998). Description of the ntu system used for MET-2. *Proceedings of the 7th Message Understanding Conference*. Fairfax, Virginia: ACL.
6. Mansouri, A., Affendey, L. S., Mamat, A. (2008). Named entity recognition approaches. *International Journal of Computer Science and Network Security, 8(2),* 339–344.
7. Morwal, S., Jahan, N., Chopra, D. (2012). Named entity recognition using hidden markov model (HMM). *International Journal on Natural Language Computing, 1(4),* 15–23. DOI 10.5121/ijnlc.2012.1402.

8. Lin, X. D., Peng, H., Liu, B. (2006). Chinese named entity recognition using support vector machines. *Proceedings of the International Conference on Machine Learning and Cybernetics,* pp. 4216–4220. Guangzhou, China: IEEE.
9. Lafferty, J., McCallum, A., Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning,* pp. 282–289. San Francisco, CA, USA: Morgan Kaufmann.
10. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. et al. (2019). Text classification algorithms: A survey. *Information-An International Interdisciplinary Journal, 10(4),* 150. DOI 10.3390/info10040150.
11. Liu, Y., Zhang, J. (2018). Deep learning in machine translation. In: Deng, L., Liu, Y. (Eds.), *Deep learning in natural language processing,* pp. 147–183. Singapore: Springer.
12. Gao, H., Huang, W., Duan, Y. (2021). The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments: A QoS prediction perspective. *ACM Transactions on Internet Technology, 21(1),* 1–23. DOI 10.1145/3391198.
13. Zhang, H., Hennig, L., Alt, C., Hu, C., Meng, Y. et al. (2020). Bootstrapping named entity recognition in e-commerce with positive unlabeled learning. *Proceedings of the 3rd International Workshop on E-Commerce and NLP,* pp. 1–6. Seattle, WA, USA: Association for Computational Linguistics.
14. Chen, M., Shen, H., Huang, Z., Luo, X., Yin, J. (2020). Towards accurate search for e-commerce in steel industry: A knowledge-graph-based approach. *Proceedings of the 16th International Conference on Collaborative Computing: Networking, Applications, and Worksharing,* pp. 3–18. Shanghai, China: Springer.
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. (2017). Attention is all you need. *Proceedings of the 31st Annual Conference on Neural Information Processing Systems,* pp. 6000–6010. Long Beach, California, USA: NILP.
16. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.
17. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pp. 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.
18. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X. et al. (2019). Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.
19. Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L. et al. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics, 8,* 64–77. DOI 10.1162/tacl_a_00300.
20. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M. et al. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.
21. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. et al. (2020). Albert: A lite bert for self-supervised learning of language representations. *Proceedings of the 8th International Conference on Learning Representations*. Addis Ababa, Ethiopia: OpenReview.net.
22. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision,* pp. 2980–2988. Venice, Italy: IEEE Computer Society.
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2818–2826. Las Vegas, NV, USA: IEEE Computer Society.
24. Yadav, V., Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. *Proceedings of the 27th International Conference on Computational Linguistics,* pp. 2145–2158. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

25. Putra, H. S., Priatmadji, F. S., Mahendra, R. (2020). Semi-supervised named-entity recognition for product attribute extraction in book domain. *Proceedings of the 22nd International Conference on Asia-Pacific Digital Libraries,* pp. 43–51. Kyoto, Japan: Springer.
26. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Proceedings of the 27th Annual Conference on Neural Information Processing Systems,* pp. 3111–3119. Nevada, USA: Lake Tahoe.
27. Ma, J., Ganchev, K., Weiss, D. (2018). State-of-the-art chinese word segmentation with BI-LSTMS. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pp. 4902–4908. Brussels, Belgium: Association for Computational Linguistics.
28. Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics, 6(7),* 483–495. DOI 10.1162/tacl_a_00034.
29. Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5(2),* 157–166. DOI 10.1109/72.279181.
30. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9(8),* 1735–1780. DOI 10.1162/neco.1997.9.8.1735.
31. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F. et al. (2014). Leag RNN rning phrase representations usinencoder-decoder for statistical machine translation. *Proceedings of the Empirical Methods in Natural Language Processing,* pp. 1724–1734. Doha, Qatar: Association for Computational Linguistics.
32. Zhang, Y., Yang, J. (2018). Chinese ner using lattice LSTM. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,* pp. 1554–1564. Melbourne, Australia: Association for Computational Linguistics.
33. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pp. 260–270. San Diego, California, USA: The Association for Computational Linguistics.
34. Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L. et al. (2018). An attention-based biLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics, 34(8),* 1381–1388. DOI 10.1093/bioinformatics/btx761.
35. Wu, F., Liu, J., Wu, C., Huang, Y., Xie, X. (2019). Neural chinese named entity recognition via CNN-LSTM-CRF and joint training with word segmentation. *Proceedings of The World Wide Web Conference,* pp. 3342–3348. San Francisco, USA: Association for Computing Machinery.
36. Liu, L., Shang, J., Ren, X., Xu, F., Gui, H. et al. (2018). Empower sequence labeling with task-aware neural language model. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence,* pp. 5253–5260. New Orleans, Louisiana, USA: AAAI Press.
37. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C. et al. (2018). Deep contextualized word representations. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pp. 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.
38. Howard, J., Ruder, S. (2018). Universal language model fine-tuning for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,* pp. 328–339. Melbourne, Australia: Association for Computational Linguistics.
39. Luo, X., Sun, J., Wang, L., Wang, W., Zhao, W. et al. (2018). Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy. *IEEE Transactions on Industrial Informatics, 14(11),* 4963–4971. DOI 10.1109/TII.2018.2854549.
40. Luo, X., Li, J., Chen, M., Yang, X., Li, X. (2021). Ophthalmic diseases detection via deep learning with a novel mixture loss function. *IEEE Journal of Biomedical and Health Informatics, 25(9).* DOI 10.1109/JBHI.2021.3083605.

41. Sun, J., Luo, X., Gao, H., Wang, W., Gao, Y. et al. (2020). Categorizing malware via a Word2Vec-based temporal convolutional network scheme. *Journal of Cloud Computing, 9(1),* 53. DOI 10.1186/s13677-020-00200-y.
42. Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). Smote: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. DOI 10.1613/jair.953.
43. He, H., Bai, Y., Garcia, E. A., Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the IEEE International Joint Conference on Neural Networks,* pp. 1322–1328. Hong Kong, China: IEEE.
44. Tkachenko, A., Petmanson, T., Laur, S. (2013). Named entity recognition in estonian. *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing,* pp. 78–83. Sofia, Bulgaria: Association for Computational Linguistics.
45. Konkol, M., Konopik, M. (2015). Segment representations in named entity recognition. *Proceedings of the 18th International Conference on Text, Speech, and Dialogue,* pp. 61–70. Pilsen, Czech Republic: Springer.
46. Malik, M. K., Sarwar, S. M. (2016). Named entity recognition system for postpositional languages: Urdu as a case study. *International Journal of Advanced Computer Science and Applications, 7(10),* 141–147. DOI 10.14569/IJACSA.2016.071019.
47. Mozharova, V., Loukachevitch, N. (2016). Two-stage approach in russian named entity recognition. *Proceedings of the International FRUCT Conference on Intelligence, Social Media and Web,* pp. 1–6. Petersburg, Russia: IEEE.
48. Alshammari, N., Alanazi, S. (2020). The impact of using different annotation schemes on named entity recognition. *Egyptian Informatics Journal, 8(1),* 37736. DOI 10.1016/j.eij.2020.10.004.