Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(4): 3553–3566.
DOI: 10.3934/mbe.2020200
Received: 31 December 2019
Accepted: 20 April 2020
Published: 12 May 2020

Research article

Fast and effective biomedical named entity recognition using temporal convolutional network with conditional random field

Chao Che[1,*], Chengjie Zhou[1], Hanyu Zhao[1], Bo Jin[2], and Zhan Gao[3]

[1] Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, Dalian University, Dalian 116622, China
[2] School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian 116024, China
[3] Beijing Haoyisheng Cloud Hospital Management Technology Ltd. Beijing 100120, China

* **Correspondence:** Tel: +86-041187402046; Email: chechao101@163.com.

Abstract: Biomedical named entity recognition (Bio-NER) is the prerequisite for mining knowledge from biomedical texts. The state-of-the-art models for Bio-NER are mostly based on bidirectional long short-term memory (BiLSTM) and bidirectional encoder representations from transformers (BERT) models. However, both BiLSTM and BERT models are extremely computationally intensive. To this end, this paper proposes a temporal convolutional network (TCN) with a conditional random field (TCN-CRF) layer for Bio-NER. The model uses TCN to extract features, which are then decoded by the CRF to obtain the final result. We improve the original TCN model by fusing the features extracted by convolution kernel with different sizes to enhance the performance of Bio-NER. We compared our model with five deep learning models on the GENIA and CoNLL-2003 datasets. The experimental results show that our model can achieve comparative performance with much less training time. The implemented code has been made available to the research community.

Keywords: biomedical named entity recognition; temporal convolutional network; conditional random field

1. Introduction

Named entity recognition (NER) aims to identify the entities from text and classifying them into predefined entity types. NER not only is a key initial step of information extraction but also plays a crucial role in many other natural language processing (NLP) tasks. Biomedical named entity recognition (Bio-NER) is the application of NER in the biomedical field and it recognizes biomedical entities from texts, such as gene, protein, drug and disease names. Bio-NER is a fundamental task in the biomedical knowledge discovery from texts. However, Bio-NER is particularly difficult because

biomedical entities have large numbers of synonyms, often use abbreviations, and include letters, symbols, and punctuation. Many studies have been conducted on these problems.

With the successful application of deep learning models in NLP, Recurrent neutral network (RNN) model [1–3] and transformer-based models [28] are the most common used models in Bio-NER. However, since RNN has a hidden layer to store the previous state information, and each unit needs the information of the previous unit as the input, RNN is computationally intensive and cannot use massive parallel processing to increase computational speed. Transformer-based models such as BioBERT [28] achieve the state-of-the-art performance. However, they require more computation than RNN due to huge number of parameters. To this end, we combine Temporal Convolutional Network [4] with a Conditional Random Field (TCN-CRF) layer for Bio-NER. Compared to RNN, TCN is more computationally efficient due to its simple structure and can be calculated in parallel when dealing with large-scale data. In TCN-CRF, the CRF layer can better extract the dependencies between labels of biomedical named entities. We also improve original TCN model by combining the features extracted by the convolution kernels with different sizes, which can effectively enhance the feature extraction ability for complex biomedical named entities. In summary, the main contribution of the paper is as follows: (1) we proposed a TCN-CRF model for Bio-NER task, which greatly improve the computation efficiency over BiLSTM and BERT models. (2) we integrated convolution kernels with different kernel sizes in TCN to extract the features, which can enhance the performance of Bio-NER task. (3) we put the CRF layer on the top of TCN to capture the dependencies between labels in the named entities. This significantly enhances the performance of NER compared to the TCN model. The source code of our model is available at https://github.com/Nickname1230/TCN-CRF-NER.

2. Related work

NER tasks are typically considered as sequential annotation tasks, and are mainly divided into statistical machine learning models and neural network-based deep learning models. Common statistical machine learning models include the hidden markov model (HMM) [5] and conditional random field (CRF) model [6]. Amongst them, the CRF model is widely used in various NER tasks with supervised learning [7–10], and has achieved better results. However, these methods are difficult to design, and have poor adaptability.

In recent years, deep learning models based on neural networks have made significant breakthroughs in performing various NLP tasks. Compared with traditional machine learning methods, the neural network model can automatically extract features and carry out end-to-end training; thus, the neural network model can achieve better results when performing NER. Collobert et al. [11] proposed a NER model based on a neural network. The model uses a CNN to extract features and fuse other linguistic features, such as part of speech tagging, even when only using word-level representation. Based on Collobert's work, Yao et al. [12] presented a multiple-layer neural network based on CNN and applied it on biomedical NER task. Wu et al. [13] used a convolution layer to generate global features represented by multiple global hidden nodes, and then input both the local and global features to standard affine networks to identify named entities in clinical texts. However, these neural network models do not consider the correlation between sequences. Therefore, in recent years, the use of RNNs has become the main research trend with regard to NER. Huang et al. [14] utilized a BiLSTM-CRF model that combines

spelling, part of speech, context, and other features to improve the model's NER performance. Based on the work of Huang et al., Lample et al. [15] further improved the performance of the model. Chiu et al. [16] introduced a BiLSTM-CNN model to further improve NER performance. The model uses a hybrid structure consisting of a BiLSTM and a character-level CNN to automatically acquire word-level features, capitals, dictionaries, and character-level features. RNN model and its variants LSTM were also applied to Bio-NER. Li et al. [17] proposed extended RNN for Bio-NER, which considers the predicted information from the prior node and external context information. Gridach [18] employed a BiLSTM-CRF model for Bio-NER and used word- and character-level representations as input. The combination of word embeddings and character-level representation helped improve the accuracy of the Bio-NER tasks. Qiu et al. [19] proposed a model named ID-CNN-CRF which feed the vector into the residual network and the final CRF layer to improve the model performance. Jiang et al. [20] employed BiLSTM networks with a CRF layer for Bio-NER. The model leverages both the pre-training and fine-tuning word embeddings as input, and inserts the sentence-level reading control gate into the network to obtain more abundant contextual information. Recently pre-trained model BERT [27] have achieved state-of-the-art performance in many fields of NLP. Lee et al. [28] adapted BERT for biomedical field and presented BioBERT (bidirectional encoder representations from transformers for biomedical Text Mining), which has almost the same architecture with BERT but is pre-trained on large-scale biomedical corpora. BioBERT largely outperforms BERT and previous state-of-the-art models in many biomedical text mining tasks including BioNER.

3. Model

The TCN-CRF model consists of three layers, which are the feature representation layer, TCN layer, and CRF layer. The overall architecture of the model is shown as Figure 1. The feature representation layer is mainly composed of a word vector layer and a character vector layer. The word vector layer and the character vector layer receive words and characters as inputs, respectively, and map discrete one-hot representations into their respective successively dense low-dimensional feature spaces. Then, the word vector and the character-level vectors are integrated together to represent the features of a word in a semantic space. Subsequently, the fusion features are used as input to the TCN. Different features are extracted by TCN with different convolution kernels, and the final features $(h_1, h_2, \cdots, h_N)$ are used as input to the CRF layer.

3.1. Embedding layer

The embedding layer maps the input word sequence into a continuous and dense feature vector, which captures the semantic information, syntactic information, and morphological information of each word. A fixed-size dictionary D_{word} and a fixed-size character set D_{char} are defined. Given a sentence $\{W_1, W_2, ..., W_n\}$ with length N, each word in the sentence is mapped to a word representation $x_n = [r_{word}; r_{char}], r_{word} \in R_{word}, r_{char} \in R_{char}$. In the representation x_n, word vectors r_{word} capture the semantic and syntactic information of words and character-level vectors and character vectors r_{char} capture the morphological information of words. The experimental results obtained by Guanghe et al. [21] revealed that the use of pre-trained word vectors can effectively improve the performance of the model because they contain context semantic information, and can thus achieve better NER results. However, traditional word vector pre-trained models, such as Word2Vec. [22], only focus on

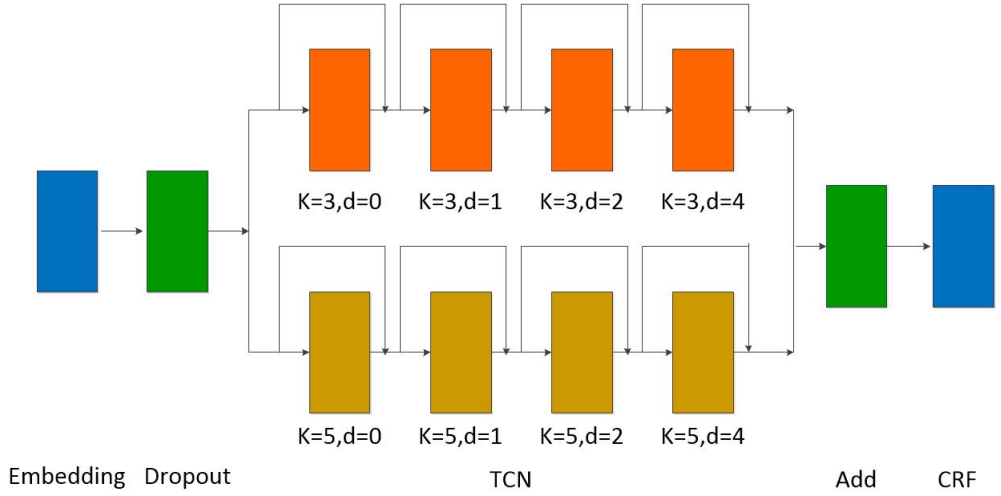

Figure 1. The architecture of the TCN-CRF model . The structure of the dilated causal convolution of TCN with dilation factors d = 0, 1, 2, 4 and kernel size of 3 or 5.

context information in the window, and cannot obtain global information. Hence, in this study, we used the GloVe. [23] pre-trained word vector model, which can obtain the global text information by combining the co-occurrence information of words on the basis of Word2Vec. In summary, this study constructed the feature representation layer using the GloVe word vector and character-level vector.

3.2. TCN layer

TCN is a member of the convolutional neural network family, and is primarily used to handle sequence problems. By using causal convolution [23], the TCN guarantees that the prediction of the previous time step will not use future information, because the output y_t of time step t only needs to be derived from the convolution operation on $t-1$ and the previous time step.

$$y_t = \prod_{t=1}^{T} P(x_t|x_1 x_2 ... x_{t-1}) \tag{3.1}$$

In the sequence labeling task, the fusion of different features can better identify entities. In the convolution operation, different sizes of convolution kernels can be used to extract different features under different receptive fields. In the TCN-CRF model, we use 1×3 and 1×5 convolution kernels to extract the features of the text, and finally fuse the features extracted by two kernels.

To remember long-term sequences, the causal convolution can broaden the receptive field by a very deep network or a very big convolution kernel. Both methods make the network difficult to train due to the huge amount of computation. To solve this problem, the TCN uses dilated convolution [25] to expand the receptive field. The dilated convolution operates on the principle of adding various weights to the convolution kernel with zero value, while keeping the input unchanged. Thus, the length of the sequence increases, such that the network can observe, while the amount of calculation remains essentially unchanged. Formally, for one-dimensional input sequences $x \in R^n$ and a kernel

$f : \{0, 1, ..., k\} \rightarrow R$, dilated convolution F is calculated as follows:

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) x_{s-d \cdot i} \tag{3.2}$$

where d is the dilated coefficient, k is the kernel size, and $s - d \cdot i$ calculates which unit should be used in the upper layer. The dilated coefficient controls the number of zeros inserted between each of the two convolution kernels. The larger expansion coefficient allows the neurons at the output to characterize a wider range of input sequences, which can more effectively expand the receptive field. Therefore, when using dilated convolution, the expansion coefficient typically increases exponentially as the network depth increases. The structure of causal convolution combined with dilated convolution is shown in Figure 2.

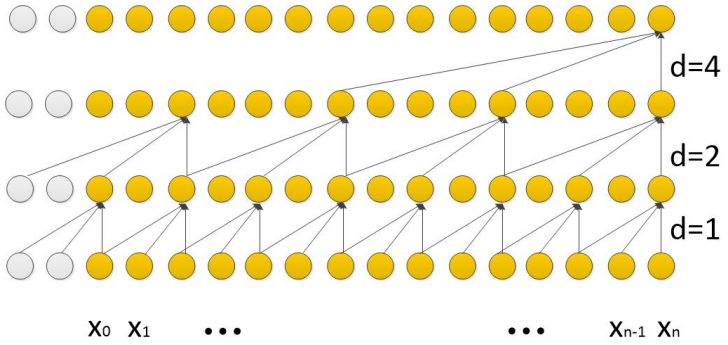

Figure 2. The structure of the dilated causal convolution of TCN.

The gradient vanishing problem is very common in deep neural networks. To address this problem, TCN adds shortcut connections in the residual network [26] to improve accuracy, as shown in Figure 3. A dilated convolution layer and the ReLU excitation function exist within the residual module of the TCN. To achieve regularization, the weight of each convolution kernel is normalized and dropout is added after each dilated convolution.

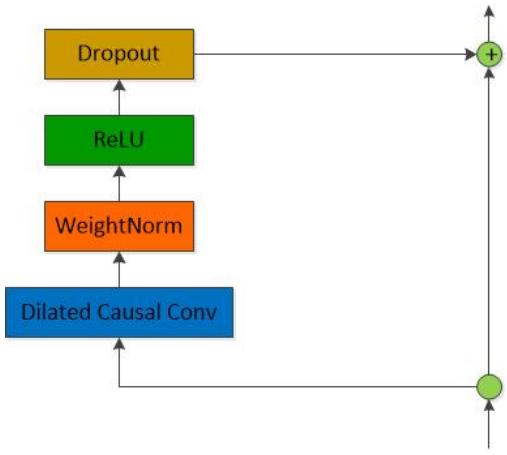

Figure 3. The structure of TCN residual block.

3.3. CRF layer

When solving the problem of sequence labeling, the softmax classifier does not consider the dependencies between labels. Thus, this paper used the CRF layer proposed by Collobert et al. [11] to consider the global information of the label sequence. The structure of CRF layer is shown as Figure 4.

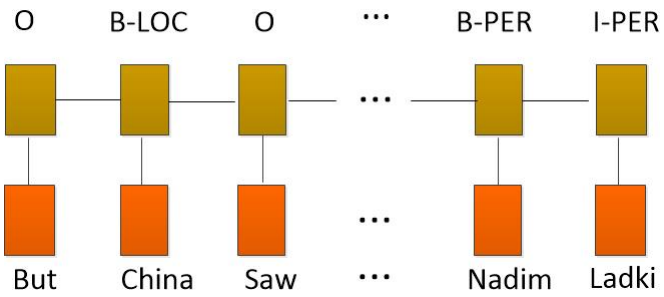

Figure 4. The structure of CRF layer.

We introduce the transfer score matrix $A_{i,j}$ for jumping from tag i to j ; y_0, y_{n+1} is the start label and the end label in the sentence, and the label type is k. Then, $A \in R^{(k+2)(k+2)}$. Let the sentence length be n, then the score matrix of the output layer is $P \in R^{(n*k)}$, and the matrix element $P_{i,j}$ indicates the output score of the ith word under label j. Given the input sentence $X = (x_1, x_2, ..., x_n)$ and the tag sequence $y = (y_1, y_2, ..., y_n)$, the total score $S(X, y)$ of the sentence X along a path of tag y is calculated as follows:

$$S(X, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \tag{3.3}$$

All possible sequence paths are normalized to produce a probability distribution for the output sequence y, as follows:

$$P(y|X) = \frac{e^{S(X,y)}}{\sum_{\bar{y} \in Y_X} e^{(S(X,\bar{y}))}} \tag{3.4}$$

The logarithmic probability of the correct tag sequence during training can be maximized as follows:

$$log(P(y^*|X)) = S(X|y^*) - log(\sum_{\bar{y} \in Y_X} e^{(S(X,\bar{y}))}) \tag{3.5}$$

As expressed in the above equation, this formula can cause the model to generate the correct tag sequence. In the decoding phase, the sequence with the highest total score is predicted to be the most optimal sequence, as follows:

$$y^* = argmaxS(X, \bar{y}) \tag{3.6}$$

The Viterbi algorithm is used to solve the optimal sequence in the prediction stage.

4. Experiments

4.1. Data

In this study, we performed TCN-CRF model for Bio-NER on the GENIA* corpus. To further investigate the effectiveness of the improvement on convolution kernels and test computational efficiency of the TCN-CRF model, we also run test on the CoNLL-2003† dataset, which is a NER dataset of general field.

The GENIA corpus is the collection of biomedical texts compiled and annotated for the GENIA project. The GENIA corpus includes five types of entities: Protein, DNA, RNA, cell Type, and cell line. The number of entries of each type is shown in Table 1. To avoid test corpus inconsistencies, the training set and test set were predefined. The training set consists 51,301 entities, while the test set has 8662 entities.

Table 1. Number of entities per type in GENIA.

	Protein	DNA	RNA	Cell Type	Cell Line	Total
Training Set	30269	9533	951	6718	3830	51301
Test Set	5067	1056	118	1921	500	8662

The CoNLL-2003 corpus, which was obtained from the Reuters news, was designed for language-independent NER task. The corpus contains four types of entries: Personal names (PER), location names (LOC), organization names (ORG), and other entities (MISC). The dataset was divided into the training set, verification set, and test set. The number of entities for four types in different datasets is shown in Table 2.

Table 2. Number of entities per type in CoNLL-2003.

	PER	LOC	ORG	MISC	Total
Training Set	6600	7140	6321	3438	23499
Validation Set	1842	1837	1341	922	5942
Test Set	1617	1668	1661	702	5648

We used precision (P), recall (R) and F_1 to evaluate the NER results. Precision is the percentage of correctly recognized entities among the entities your model recognized. Recall is the percentage of correctly recognized entities among total entities. F_1 is the harmonic mean of precision and recall, which is calculated as follows:

$$F_1 = \frac{2 * P * R}{P + R} \tag{4.1}$$

where P is the precision and R is the recall.

*https://www.geniaproject.org/home
†https://www.clips.uantwerpen.be/conll2003/ner

4.2. The effectiveness of the CRF layer

To validate the effectiveness of the CRF layer, we compare TCN-CRF model with the TCN model using softmax layer instead of CRF layer. The results of two models on the GENIA and CoNLL-2003 dataset are reported in Table 3. The experimental results show that TCN-CRF outperforms TCN with a large margin especially on the GENIA dataset. The CRF layer considers the relationship between the entity labels, using the CRF layer can make sure the model does not output the invalid entity labels and significantly boost the performance of NER tasks. Biomedical named entities usually have more words and modifiers. This allows the CRF layer to make better use of the dependency information of adjacent labels and obtain better results on Bio-NER.

Table 3. Comparison of TCN model and TCN-CRF model on two datasets.

	GENIA			CoNLL-2003		
	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
TCN [34]	57.81	59.10	58.45	86.32	84.56	85.43
TCN-CRF	73.46	79.61	76.41	91.48	91.37	91.42

4.3. Comparison with other deep learning models

Tables 4 and 5 compare the TCN-CRF model with five popular sequence labeling models, i.e., CNN-CRF, BiLSTM-CRF, ID-CNN-CRF, BERT and BioBERT, on the GENIA and CoNLL-2003 datasets. Because BioBERT is a domain-specific model pretrained on biomedical data, we did not implement it on the CoNLL-2003 dataset.

Table 4. Comparison with other deep learning models on the GENIA.

Model	Feature	F_1(%)
CNN-CRF [30]	Word2Vec, POS	71.01
Multi-layer BiLSTM [29]	Word Embedding	73.80
BiLSTM-CRF [31]	Word, Character	74.70
BERT [27]	–	72.49
BioBERT [28]	–	77.15
TCN-CRF	GLoVe, Character	76.45

Table 5. Comparison with other deep learning models on the CoNLL-2003.

Model	Feature	F_1(%)
CNN-CRF [11]	Word indices, POS	89.86
LSTM-CRF [14]	SENNA, Spelling, N-gram, Gazetteer	90.10
ID-CNN-CRF [32]	SENNA, Word Shape	90.65
BiLSTM-CNN [16]	SENNA, Character, Capitalization, Lexicons	91.62
BiLSTM-CRF [33]	GLoVe, Character	91.26
BERT [27]	–	92.80
TCN-CRF	GLoVe, Character	91.42

In Tables 4 and 5, we can observe that BioBERT achieves the best performance on GENIA and BERT performs best on CoNLL-2003. The performance of the TCN-CRF model is in line with results of the models that perform the best on the evaluation datasets. The receptive field of the CNN model is small and fixed, and cannot effectively capture context information, which is used to predict the label in NER task. The ID-CNN model expands the receptive field with help of the dilated convolution. However, the traditional convolution operation cannot effectively obtain the timing information. LSTM can effectively solve the problem of limited scope of context information accessed by RNN through gate structure. However, due to the existence of forget gate, LSTM will selectively discard the information in the past when updating cell state. Therefore, when the current time step is used to predict the model based on LSTM, all the original information of the past time step cannot be retrieved quickly, which will lead to the performance degradation of the model to some extent. Although the BiLSTM model can obtain the context information of the sequence by the information transmission mechanism of the gate, the back propagation along the time axis often leads to the gradient disappearance or gradient explosion. The TCN-CRF model expands the receptive field by combining causal convolution and dilated convolution, and extracts the time series information well. In addition, the back propagation path of TCN and the time direction of the sequence are different, which avoids the gradient explosion and gradient disappearance problems that often occur with RNN series models. Therefore, TCN-CRF is superior to most other deep learning models. Additionally, considering it is easy to change the size of the receptive field of TCN-CRF, we use different sizes of convolution kernels to extract the text features and fuse the features to further improve the performance of NER.

4.4. Comparison of training time

We compared the computational efficiency of TCN-CRF, LSTM-CRF, BiLSTM-CRF, BERT and BioBERT on the GENIA and CoNLL-2003 datasets. The training time of 5 models on two datasets is shown in Figures 5 and 6, respectively. BioBERT was not implemented on GENIA since it is a specific-domain model. The experiment was run on a NVIDIA 1080Ti GPU.

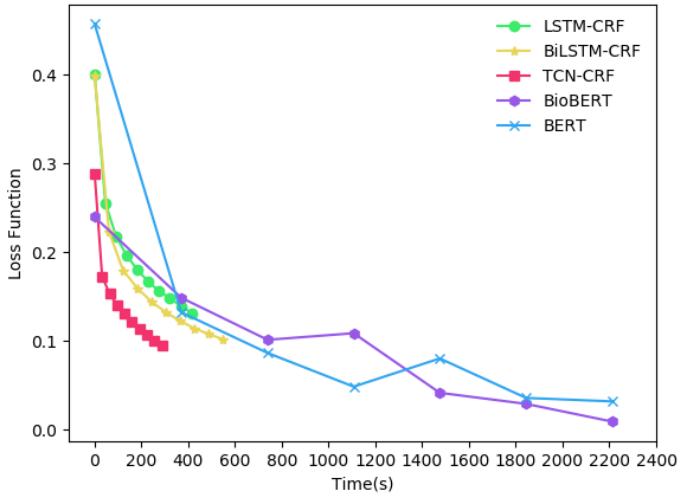

Figure 5. Comparison of training time of five deep learning models on the GENIA dataset.

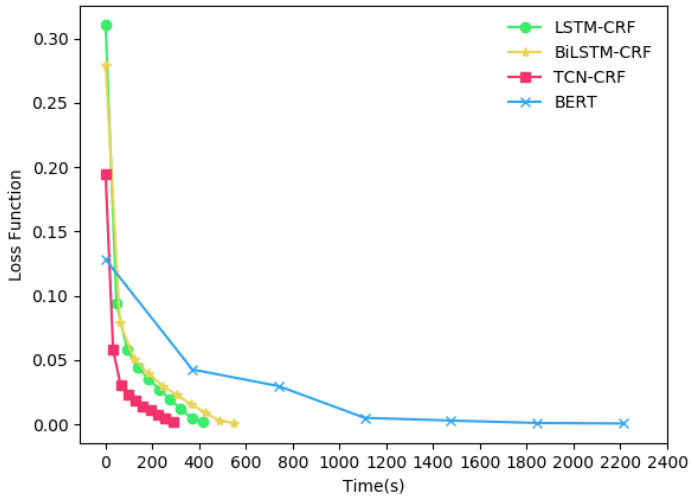

Figure 6. Comparison of training time of four deep learning models on the CoNLL-2003 dataset.

For both the GENIA and CoNLL-2003 datasets, the convergence time of each iteration of the TCN model is obviously less than those of LSTM-CRF, BiLSTM-CRF and BERT. The training time of TCN-CRF is approximately 1/2 of LSTM-CRF, 1/3 of BiLSTM-CRF, and 1/7 of BERT. As the variations of RNN, LSTM and BiLSTM use the output of the unit as the input of the next unit, so the training process of the model can only be performed serially. The training speed of BERT is very slow due to the large number of parameters. As a member of CNN, TCN is calculated layer by layer, that is, multiple convolution kernels can be calculated simultaneously, not serially in time sequence. Because of the large-scale parallel processing, the computing efficiency of TCN is much higher than that of other networks such as LSTM.

4.5. Effect of using convolution kernels with different sizes on TCN

In TCN-CRF, we extracted the features in different receptive fields convolution kernels with different sizes, which are integrated and input into the CRF layer. To verify the effectiveness of kernels fusion, we conduct an ablation study. We run TCN-CRF using three convolution kernels, namely, 1 × 3 kernel, 1 × 5 kernel and a kernel integrating 1 × 3 kernel and 1 × 5 kernel. The F_1 of TCN-CRF using different convolution kernels is shown in Table 6.

Table 6. The F_1 of TCN-CRF using different convolution kernels on two datasets.

Convolution kernel	GENIA(%)	CoNLL-2003(%)
size = 3	75.66	90.66
size = 5	75.14	90.07
size = 3 and 5	76.45	91.42

Table 6 shows that using convolution kernels of different sizes is superior to using only one size of convolution kernel. This because the convolution networks fusing different sizes of kernels can extract features of different receptive fields. When using smaller convolution kernels, the extracted features will have finer granularity, and we can obtain more global information using larger receptive field convolution kernels. The fusion of the convolution kernels of two sizes makes the features extracted by the model more diverse. When a single size convolution kernel is used, the elements in the hidden layer of the sequence that are located at the dilated convolution positions are not calculated, which will lose the continuity of the information. The fusion of multiple convolution kernels of different sizes can reduce the impact of hidden layer element discontinuities on model performance.

4.6. The effect of Dropout layer

To investigate how to use dropout in TCN-CRF, we implemented three types of dropout: (1) not using the dropout layer; (2) using the dropout only after the input layer; (3) dropout is used in both input layer and the residual structure. The F_1 of TCN-CRF using different dropout is shown in Table 7.

Table 7. The F_1 of TCN-CRF using different dropout.

Inputs	Residual block	CoNLL-2003(%)	GENIA(%)
–	–	87.20	71.23
+	–	88.11	73.79
–	+	90.80	74.80
+	+	91.40	76.45

Our experiments show that using dropout can effectively improve the performance of deep learning

model. When using dropout, two neurons do not necessarily appear in a network every time. In this way, the updating of weights no longer depends on the joint action of hidden nodes with fixed relationships, so that some patterns can still be learned from other information in the case of losing certain information and the robustness of the model is improved. From the perspective of model integration, we can treat the residual network of 3-layer as a binary tree with 8 nodes, which integrates the output of the 8 nodes, and the final output is the integration of these 8 nodes. Using dropout to randomly delete some nodes of the residual network will generate more network structures, which can effectively prevent overfitting and improve the performance of the model.

5. Conclusions

In this paper, we propose a temporal convolutional network-conditional random field (TCN-CRF) model for Bio-NER. We put CRF layer on the top of TCN to capture the dependencies between labels and improve the feature extraction ability of TCN by integrating convolution kernels with different kernel sizes. We evaluate our method on the GENIA and CoNLL-2003 datasets. TCN-CRF greatly reduces training time compared with other deep learning models while achieving comparative performance. Although TCN-CRF is superior to traditional neural network models, the performance of TCN in transfer learning requires further improvement. Because the historical information required for model prediction is different in different data sets, the size of the receptive field cannot be effectively controlled. In future work, we will design an adaptive receptive field adjustment mechanism to better obtain the sequence information.

Acknowledgments

The work is supported by the Program of National Key R&D Program of China (2018YFC0116800), the National Natural Science Foundation of China (No. 61772110), the Guidance Program of Liaoning Natural Science Foundation (No. 2019-ZD-0569), the CERNET Innovation Project (NGII20170711), the Introducing Talents of Discipline to Universities (Plan 111) (No. B20070).

Conflict of interest

The authors declare that they have no competing interests.

References

1. D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by backpropagating errors, *Nature*, **323** (1986), 533–536.
2. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780.
3. Q. Wang, Y. Zhou, T. Ruan, Y. Xia, D. Gao, P. He, Incorporating dictionaries into deep neural networks for the Chinese clinical named entity recognition, *J. Biomed. Inform.*, **92** (2019), 103133.
4. S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, *arXiv preprint arXiv*, **2018** (2018), 1803.01271.

5. G. Zhou, J. Su, *Named entity recognition using an HMM-based chunk tagger*, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002. Available from: https://dl.acm.org/doi/10.3115/1073083.1073163.
6. J. Lafferty, A. Mccallum, F. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, Proceedings of the 18th International Conference on Machine Learning, 2001. Available from: https://repository.upenn.edu/cis_papers/159/.
7. D. Lin, X. Wu, *Phrase clustering for discriminative learning*, The International Joint Conference on Natural Language Processing, 1997. Available from: https://dl.acm.org/doi/10.5555/1690219.1690290.
8. A. Passos, V. Kumar, A. Mccallum, Lexicon infused phrase embeddings for named entity resolution, *arXiv preprint arXiv*, **2014** (2014), 1404.5367
9. G. Luo, X. Huang, C. Lin, Z. Nie, Joint entity recognition and disambiguation, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015. Available from: https://www.aclweb.org/anthology/.
10. B. Wang, Q. Zhang, X. Wei, Tabu Variable Neighborhood Search for Designing DNA Barcodes, *IEEE Trans. Nanobiosc.*, **19** (2020), 127–131.
11. R. Collobert, J. Weston, L. Bottou, Nature language processing (almost) from scratch. *J. Mach. Learn Res*., **12** (2011), 2493–2537.
12. L. Yao, H. Liu, Y. Liu, D. Huang, Biomedical named entity recognition based on deep neutral network, *Int. J. Hybrid Inform.*, **8** (2015), 279–288.
13. Y. Wu, M. Jiang, J. Lei, H. Xu, Named entity recognition in Chinese clinical text using deep neural network, *Stud. Health Technol.*, **216** (2015), 216–624.
14. Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging,*arXiv preprint arXiv*, **2015** (2015), 1508.01991.
15. G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics, *arXiv preprint arXiv*, **2016** (2016), 1603.01360.
16. J. P. C. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, *Trans. Assoc. Comput. Linguist.*, **4** (2016), 357–370.
17. L. Li, L. Jin, Z. Jiang, *Biomedical Named Entity Recognition Based on Extended Recurrent Neural Networks*, IEEE International Conference on Bioinfonnatics and Biomedicine, 2015. Available from: https://ieeexplore.ieee.org/.
18. M. Gridach, Character-Level Neural Network for Biomedical Named Entity Recognition, *J. Biomed. Inform.*, **70** (2017), 85–91.
19. J. Qiu, Q. Wang, Y. Zhou, *Fast and Accurate Recognition of Chinese Clinical Named Entities with Residual Dilated Convolutions*, 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2018. Available from: https://ieeexplore.ieee.org/.
20. L. Li, Y. Jiang, *Biomedical Named Entity Recognition Based on the Two Channels and Sentence-level Reading Control Conditioned LSTM-CRF*, 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2017. Available from: https://ieeexplore.ieee.org/.

21. G. Lin, S. Zhang, H. Lin, Named entity identification based on fine-grained word representation, *J. Chin. Inform. Process.*, **32** (2018), 62–72.
22. T. Mikolov, K. Chen, G. S. Corrado, Efficient estimation of word representations in vector space, *arXiv preprint arXiv*, **2013** (2013), 1301.3781.
23. J. Pennington, R. Socher, C. D. Manning, *Glove: Global vectors for word representation*, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014. Available from: https://www.aclweb.org/anthology/.
24. T. J. Brazil, Causal-convolution-a new method for the transient analysis of linear systems at microwave frequencies, *IEEE Trans. Microwave Theory Tech.*, **43** (1995), 315–323.
25. A. V. Den Oord, S. Dieleman, H. Zen, WaveNet: A generative model for raw audio, *arXiv preprint arXiv*, **2016** (2016), 1609.03499.
26. K. He, X. Zhang, S. Ren, *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016. Available from: http://cvpr2016.thecvf.com/.
27. J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv preprint arXiv*, **2018** (2018), 1810.04805.
28. J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, J. Kang, BioBERT: A pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics*, **36** (2020), 1234–1240.
29. A. Katiyar, C. Cardie, *Nested named entity recognition revisited*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018. Available from: https://www.aclweb.org/anthology/.
30. L. Yao, H. Liu, Y. Liu, X. Li, M. W. Anwar, Biomedical named entity recognition based on deep neutral network, *Int. J. Hybrid Inf. Technol.*, **8** (2015), 279–288.
31. M. Ju, M. Miwa, S. Ananiadou, *A neural layered model for nested named entity recognition*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018. Available from: https://www.aclweb.org/anthology/.
32. E. Strubell, P. Verga, D. Belanger, A. McCallum, Fast and accurate entity recognition with iterated dilated convolutions, *arXiv preprint arXiv*, **2017** (2017), 1702.02098.
33. X. Ma, E. Hovy, End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, *arXiv preprint arXiv*, **2016** (2016), 1603.01354.
34. H. Zhao, C. Che, B. Jin, A Viral Protein Identifying Framework Based on Temporal Convolutional Network, *Math. Biosci. Eng.*, **16** (2019), 1709–1717.

AIMS Press