American Journal of Computer Science and Technology
2023; 6(2): 67-73
http://www.sciencepublishinggroup.com/j/ajcst
doi: 10.11648/j.ajcst.20230602.13
ISSN: 2640-0111 (Print); ISSN: 2640-012X (Online)

Chinese NER with Softlexion and Residual Gated CNNs

Zhang Yinglin, Liu Changhui, Huang Shufen

College of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan, China

Email address:
945798576@qq.com (Zhang Yinglin), lch198@sohu.com (Liu Changhui)

To cite this article:
Zhang Yinglin, Liu Changhui, Huang Shufen. Chinese NER with Softlexion and Residual Gated CNNs. *American Journal of Computer Science and Technology.* Vol. 6, No. 2, 2023, pp. 67-73. doi: 10.11648/j.ajcst.20230602.13

Received: April 20, 2023; **Accepted**: May 18, 2023; **Published**: May 29, 2023

Abstract: The increment of accuracy and speed on Named Entity Recognition (NER), a key task in natural language processing, can further enhance downstream tasks. The method of residual gated convolution and attention mechanism is proposed to address the problem of insufficient recognition of nested entities and ambiguous entities by convolutional layers in the absence of context. It emphasizes local continuous features fusion to global ones to better obtain contextual semantic information in the stacked convolutional layer. Moreover, the optimized embedding layer with fusing character and lexical information by introducing a dictionary combines with a pre-trained BERT model containing a priori semantic effects, and the decoding layer in an entity-level method to alleviate the problem of nested entities and ambiguous entities in long-sequence text. In order to reduce abundant parameters of Bert model, during the training process, only the residual gated convolutional layer is iterated after fixing Bert layer parameters. After experiments on MSRA corpus, the result of entity recognition task in BERT-softlexion-RGCNN-GP model outperforms other models, with an F1 value of 94.96%, and the training speed is also better than that of the bidirectional LSTM model. Our model not only maintains a more efficient training speed but also recognizes Chinese entities more precisely, which is of practical value for fields required accuracy and speed.

Keywords: NER, BERT, Lexion, Residual Gated CNNs

1. Introduction

Named Entity Recognition are the recognition of relevant entity types from natural text and the determination of entity labels. There is identifiable information in the financial domain such as companies, brands, and legal entities, and as in the medical field are diseases, symptoms, and patient ages entities. As a fundamental task of natural language processing, accurate NER tasks can effectively improve the completion of downstream tasks, for instance, knowledge graphs, automatic question and answer, and machine translation.

The mainstream recognition approach is to utilize the bi-directional capturing capability of BiLSTM for long text sequences, build a suitable model structure, and improve the probability calculation and the representation of embedding layers to improve the accuracy of named entity recognition. With the growth of text sequence length as well as the number of model parameters, some researchers have also adopted convolutional layers as the backbone structure of recognition models. The method of reducing model training time by parallel CNN models and stacked convolutional layers capturing contextual semantics are used to improve entity recognition accuracy in long utterances, however, pure deep CNNs cannot solve the problem of long-distance dependency.

Therefore, we propose an incorporated residual gated convolution entity recognition model, which combines local continuous features and high-dimensional spatial semantics to selectively keep association information, and adds an attention mechanism to capture the important semantics associated with labels in the sequence:

(1) To address the inadequate grasping of contextual information by stacked CNNs, the component of residual gated convolution and attention mechanism, which fuse local features to the global and reduce invalid information input, eventually alleviate the problems of gradient disappearance in convolutional layers and semantic dependence caused by cross-layers.

(2) To address the problems of nested entities and unregistered words, a pre-trained language model combined with lexical enhancement is proposed as an embedding layer, which introduces lexical information and a priori semantics

of the large language model, to mine latent semantic information and alleviate the conflict between unregistered words and nested entities.

(3) To address the threshold problem of sequence label prediction in multiclassification datasets, Global Pointer (GP) [1] is invoked to perform label prediction of sequences with entity-level granularity to reduce the extraction error problem caused by correct sequence labeling but overly strict or lax determination conditions.

2. Related Research

NER tasks include rule-based [2, 3], machine learning-based [4-8], and deep learning-based approaches. In recent years, neural network-based deep learning models have become a hot research topic due to the limitations of manual features.

Since neural networks automatically learn and capture semantic features from the corpus, the effectiveness of entity recognition relies heavily on the representation of word embeddings. There're three types of word embedding representations in the current NER task: word-level, character-level, and hybrid representations.

Word level is an intuitive way of clause splitting and is inherited from traditional recognition tasks [8]. Nowadays, word separation is usually performed with the help of external tools such as Jieba and Hanlp to improve efficiency. Huang [10] proposed a word-level model based on LSTM-CRF [9] that effectively improves the performance of entity recognition. However, the coarse granularity at the word level leads to a significant increase in the parameters of the embedding layer but also introduces Out-of-vocabulary (OOV) problem.

The character-level representation can solve the problem of OOV and also avoid the propagation of subword errors. In the English corpus, the minimum division is character-level, and prefixes and suffixes composed of characters in words play a landmark help in the annotation. Lample [11] input the prefix and suffix extracted morphological features with word vector splicing into LSTM by modifying word embedding layer.

Hybrid representation is the fusion of multiple features as the input of a neural network. Ma [12] and Chiu [13] further optimized the input part by adding CNN (convolutional neural networks) to encode character information to capture long-range semantics, mixing character embedding and word embedding as the input of LSTM to construct contextual information. Dong [14] was inspired to bring in an assembly of Chinese character paraphernalia and combine sentiment features. Peng [15] chose a lexicon to enhance the embedding layer information and took spliced word vectors as input to a bidirectional LSTM. Lattice LSTM [16], based on the character-based model, integrated hidden lexical-level semantic information and achieved 93.18% on the MSRA corpus F1 value.

The named entity recognition task can be regarded as a sequence annotation task, in which RNNs (recurrent neural networks) are widely used. Because the BiLSTM model has a strong ability to capture contextual semantics and sequence modeling, which has achieved remarkable results in sequence labeling tasks, a series of subsequent studies [10-13] have used it as the structural basis. Nevertheless, with the growth of sequences, the long sequence modeling ability diminishes, so some studies [17, 18] used CNN as the backbone structure with higher parallelism than LSTM and addressed the problem of contextual semantic capture by deep CNN stacking. Strubell [18] proposed the use of IDCNN for named entity recognition to improve the training speed while maintaining recognition accuracy.

Overall, the convolutional layer fused with residual gated as the encoder combine with used GP as the decoding layer to better identify nested entities, by introducing annotations in the semantic part of the extracted vocabulary. In more detail, the encoder stage uses stacked dilated convolution kernels to perform parallel calculations on the entire text sequence, expand the scope of feature capture, and extract contextual high-level semantic features of sentences. Then, a residual gated unit is introduced to fuse the local context features into the global, acquire contextual semantics, and alleviate the problem of gradient disappearance caused by cross-layer propagation. In addition, to solve the problem of labeling the same entity in different contexts, a multi-head attention mechanism is introduced to extract the global features of sentences and solve the long-distance dependency problem.

3. Model

The overall structure of the entity recognition model is shown in Figure 1, and the whole model is divided into three parts: the embedding layer, the encoding layer, and the decoding layer. Among them, the embedding layer contains the pre-training model obtained a dynamic vector representation, which effectively alleviates the problem of multiple meanings at a time. Then, the vectors of the embedding layer are input to the coding layer for feature extraction, which extracts word features in convolution layer and further acquires high-dimensional lexical semantics in combination with the residual gated convolution layer. Furthermore, parameters are set to N=5, convolution kernel 3x3, and dilation set at 1,2,4,1,1 in residual gated convolution module. Finally, the decoding layer uses GP to complete the label prediction of the sequence and achieve the global optimal sequence through global normalization in combination with relative position coding.

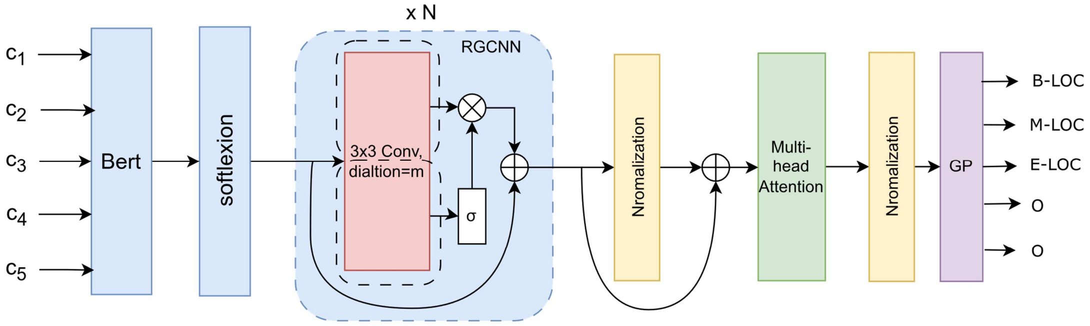

Figure 1. *Model Framework.*

3.1. Embedding Layer

The embedding layer consists of the BERT [20] model embedding layer and the vector representation of softlexion [15]. The structure of the BERT model is shown in Figure 2. which uses Transformer encoder as the basic architecture, and the input layer is the sum of word embedding, location embedding and segmentation embedding, and position embedding with temporal information, and then, label embedding that also integrated extra dictionary for softlexion part.

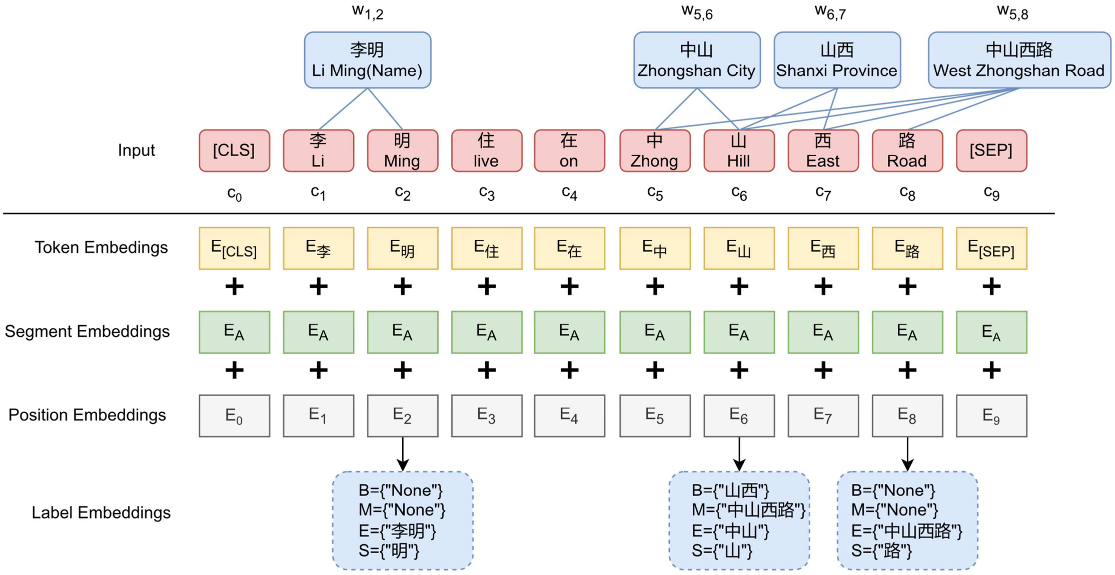

Figure 2. *Embedding Layer.*

With the avoidance of word separation, a character-level embedding-based representation is proposed to reduce the number of unregistered words. The character-level embedding layer is output to the BERT pre-training model. Word enhancement effectively alleviates the boundary recognition error problem by assigning each character to the set {B, M, E, S} and then stitching it to BERT's word vector after the whole as an embedding layer.

$$\mathrm{B}(\mathrm{c}_i) = \{\mathrm{w}_{i,k}, \forall \mathrm{w}_{i,k} \in \mathrm{L}, \mathrm{i} < \mathrm{k} \leq \mathrm{n}\} \tag{1}$$

$$\mathrm{M}(\mathrm{c}_i) = \{w_{\mathrm{j,k}}, \forall w_{\mathrm{j,k}} \in \mathrm{L}, 1 \leq \mathrm{j} < \mathrm{i} < \mathrm{k} \leq \mathrm{n}\} \tag{2}$$

$$\mathrm{E}(\mathrm{c}_i) = \{w_{\mathrm{j,i}}, \forall w_{\mathrm{j,i}} \in \mathrm{L}, 1 \leq \mathrm{j} < \mathrm{i}\} \tag{3}$$

$$\mathrm{S}(\mathrm{c}_i) = \{c_{\mathrm{i}}, \exists c_{\mathrm{i}} \in \mathrm{L}\} \tag{4}$$

Since the frequency of word is static value obtained offline, the method of counting word's frequency as a weight greatly speeds up the calculation of the weight of each word. Specifically, let $z(w)$ represent the frequency of the word w in the dictionary appearing in the statistical data, $\mathrm{e}^{w}(w)$ corresponds to the embedding of the contributing word, and the weighted representation of the word set S is as follows:

$$v^s(\mathrm{S}) = \frac{4}{Z}\sum_{w \in S} z(w)e^w(w) \tag{5}$$

The embedding of the four sets is combined into one fixed dimensional feature and added to each character representation.

$$\mathrm{e}^s(\mathrm{B}, \mathrm{M}, \mathrm{E}, \mathrm{S}) = [v^s(\mathrm{B}); v^s(\mathrm{M}); v^s(\mathrm{E}); v^s(\mathrm{S})] \tag{6}$$

$$x^c \leftarrow [x^c; \mathrm{e}^s(\mathrm{B}, \mathrm{M}, \mathrm{E}, \mathrm{S})] \tag{7}$$

3.2. RGCNN Layer

3.2.1. Residual Gated Convolution

The model is fed into a gated convolutional unit with residuals after the sequence is encoded in a pre-training layer. By adding a gated mechanism to the one-dimensional inflated convolution, the necessary high-dimensional semantic information is selectively extracted while expanding the context selection range.

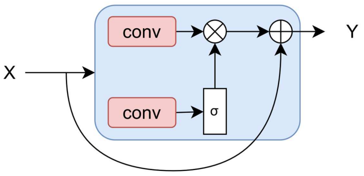

***Figure 3.** Residual Gated Linear Unit.*

The residual gated linear unit is improved from GLU (gated linear unit), based on which the residual mechanism is introduced, as in Eq.8.

$$GLU(X) = f_{w_1}(X) \otimes \sigma\left(f_{w_2}(X)\right) \tag{8}$$

f_{w_i} denotes the same convolution operation, but the weights will not be shared, where f_{w_2} uses the sigmoid activation function and f_{w_1} does the linear operation, so Eq. 9 and 10 are equivalent.

$$Y = X + f_{w_1}(X) \otimes \sigma\left(f_{w_2}(X)\right) \tag{9}$$

$$Y = X \otimes \left[1 - \sigma\left(f_{w_2}(X)\right)\right] + f_{w_1}(X) \otimes \sigma\left(f_{w_2}(X)\right) \tag{10}$$

The information passing probability of input *x* is controlled by two parts: the first part has the probability to pass directly, and the second part controls the passing probability through the gating of convolution operation, which alleviates the gradient disappearance problem by expanding the number of information transmission channels.

3.2.2. Expansion Convolution

The first application was in image domain, to expand the receptive field of the convolution kernel while keeping the size of the feature map constant, as in Figure 4. The model is a stack of four identically sized inflated convolution blocks, each containing three layers of inflated convolution with expansion widths of 1, 1, and 2. [18] covers the entire sequence more rapidly by allowing the perceptual field to grow exponentially and remaining the number of parameters.

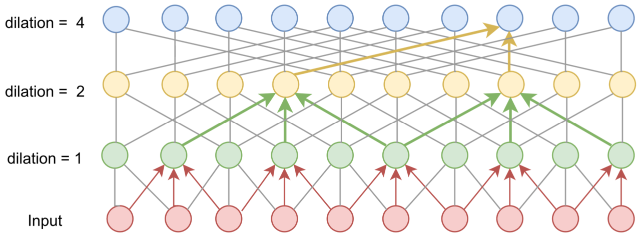

***Figure 4.** Dilated Convolution.*

3.3. Attention Mechanism

The encoder of Transformer consists of a combination of attention mechanism and feedforward neural network, as in Figure 5.

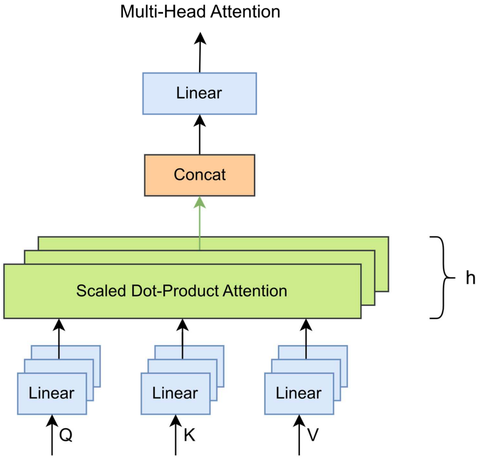

***Figure 5.** Multi-head Attention diagram.*

The attention mechanism is the core part of the encoder. After the input of the BERT model, the attention operation is performed to calculate the information related to the other vectors in each word vector.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{11}$$

After projecting Q, K, and V in different linear spaces, splices of all the attention results are calculated as in Eq. 12.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{12}$$

$$\text{Multi-head}(Q, K, V) = \text{Concat}(\text{head}_1, \cdots, \text{head}_k)W^o \tag{13}$$

Then the splicing result is connected with the input residual of the BERT layer in the normalized calculation to obtain a normal distribution result, which continues to input into the feedforward neural network, and the dimensionality reduction operation is completed through two linear transformations.

3.4. Decoding Layer

The sequence of encoded vectors obtained after encoding the input sentence *t* of length n is $[h_1, h_2, \cdots, h_n]$. The encoding vector of each token is put into two linear layers to record the query and key belonging to each entity class, respectively:

$$q_{i,\alpha} = W_{q,\alpha}h_i + b_{q,\alpha} \tag{14}$$

$$k_{i,\alpha} = W_{k,\alpha}h_i + b_{k,\alpha} \tag{15}$$

The α denotes a class of entities, and here it is equivalent to trying different q and k for various entity classes identified by $t_{[i,j]}$ of consecutive substrings, scoring the entity α:

$$s_\alpha(i,j) = q_{i,\alpha}^{\mathrm{T}} k_{j,\alpha} \tag{16}$$

Within entities scores, conversions in consecutive substrings add the rotation position code RoPE, which is a transformation matrix R_i that satisfies $R_i^T R_j = R_{j-1}$:

$$s_\alpha(i,j) = \left(R_i q_{i,\alpha}\right)^T \left(R_j k_{j,\alpha}\right) = q_{i,\alpha}^{\mathrm{T}} R_{j-i} k_{j,\alpha} \tag{17}$$

Since the final scoring function corresponds to α n(n+1)/2 class binary classification problems, in case of severe class imbalance for each type of entity candidate, the loss function uses a single-objective cross-entropy generalization of the multiclassification:

$$L(i,j) = log\left(1 + \sum_{(i,j)\epsilon P_\alpha} e^{-s_\alpha(i,j)}\right) + log(1 + \sum_{(i,j)\epsilon Q_\alpha} e^{-s_\alpha(i,j)}) \tag{18}$$

Among samples, P_α is the sum of the closing sets of all entities of type α for that sample, and Q_α is the sum of the closing sets of entities of type not α for that sample or all non-entities, considering only the combinations i ≤ j:

$$\Omega = \{(i,j) | 1 \le i \le j \le n\} \tag{19}$$

$$P_\alpha = \left\{(i,j) \middle| t_{[i,j]} \text{ is an entity of type } \alpha\right\} \tag{20}$$

$$Q_\alpha = \Omega - P_\alpha \tag{21}$$

All segments that satisfy $s_\alpha(i,j) > 0$ fragments of $t_{[i,j]}$ are considered as entity outputs of type α.

4. Experiments

4.1. Details

The evaluation indicators of the experiment are precision rate P, recall rate R, and F1 value. Dimension label of BIO from MSRA converted to BMES. The experimental environment is 1080ti, 64G memory. The parameters of the model are set as follows: Bert model adopted bert-base-chinese version, 12 heads mode. The hidden layer dimension is set to 768, batch size to 128, learning rate to 1e-3, and maximum input text length to 512. Using Adam optimizer to prevent overfitting, Dropout is set to 0.2.

4.2. Model Computing Efficiency

The calculation efficiency of the RGCNN model is in comparison with basic models on MSRA, comprising BiLSTM model commonly used in NER task, IDCNN based on expansive convolution, and GRN [19] model based on residual gated, to compare the single step time of the model processing the same batch of samples and updating the weight once.

Table 1. *Model single-step efficiency.*

Models	P	R	F1	Single step/ms
BiLSTM [10]	85.60	84.55	85.12	416
GRN [19]	91.16	88.68	89.89	189
IDCNN [18]	87.11	86.42	87.97	124
RGCNN	91.88	90.55	91.21	158

CNN-based models are generally faster to train than RNN models, and achieve higher F1 values. Among them, the speed of IDCNN is nearly 3 times faster than that of BILSTM, the single-step time of RGCNN based on expansion convolution is 2.5 times faster than that of BiLSTM, and the F1 value is 6.11% higher than which. The accuracy of the GRN model resembles that of RGCNN, but the single-step time and recall rate are not as good as that of RGCNN.

It can be seen that the CNN-based model has a significant speed advantage. The main reason is that the RNN model has to recursively obtain the global information, while CNN obtains the information by increasing the perceptual field through layer stacking, and the operations of each layer are parallel, so the speed of the model is greatly improved.

4.3. Impact of Embedding Layer on Entity Extraction

Based on the RGCNN model, a pre-training model is added to verify the influence of prior semantics on entity extraction. Character embedding, word embedding, and context embedding are respectively introduced to perform comparative experiments with the BERT model, which are 128,128 and 256, respectively. Then compare the accuracy of other vocabulary enhancement methods in the NER task.

Table 2. *Embedding layer comparison.*

Models	P	R	F1
RGCNN	88.31	86.92	87.61
RGCNN+character embedding	90.79	85.26	87.93
RGCNN+position embedding	90.21	87.45	88.80
RGCNN+context embedding	92.88	90.42	91.63
RGCNN+Bert	92.94	91.53	92.22
RGCNN+softlexion	91.36	90.48	90.92
RGCNN+Bert+softlexion	94.45	93.82	94.13

Algorithm improvement from vectors obtained from the embedding layer indicates F1 value of the embedding layer with the fused context in Table 2 is improved by 3.5% compared to using only RGCNN, verifying the improvement of contextual information for entity recognition. The overall evaluation criteria are all improved after embedding enhancement. Due to the sufficient word frequency statistics in the MRSA training set, the effect of RGCNN+softlexion differs only 1.3% from BERT. The fusion of BERT and softlexion has more improvement for entity recognition, mainly attributed to the a priori semantics of the BERT model.

4.4. Model Validation

The following comparative experiments are conducted to verify effectiveness of proposed model as shown in Table 3: 1) To verify the effect of the depth of the BERT model on the entity recognition effect, the model depths of 6, 8, and 10 layers were selected for the experiments. 2) Based on the optimal model depth, two decoding methods, CRF and GP, were used to compare the entity recognition. 3) The RGCNN+GP model and the mainstream BiLSTM+CRF model are compared.

Table 3. *Comparison of model groups.*

Group	Model	Embedding	Depth	F1
1	RGCNN+GP	BERT	6	91.73
2		BERT	8	93.42
3		BERT	10	92.57
4	RGCNN+CRF	BERT	8	92.22
5	BiLSTM+CRF	Word2vec	1	89.13

Several sets of different BERT layers demonstrate that the F1 value using the BERT pre-training model reaches the highest value of 93.42% at depth 8, the lowest value at depth 6, and the average value between depth 6 and 8 at depth 10. It indicates that appropriately deepening the number of network layers is beneficial to improve the accuracy of entity recognition, but as the model continues to deepen, the learning ability of the model decreases and causes degradation of the recognition effect.

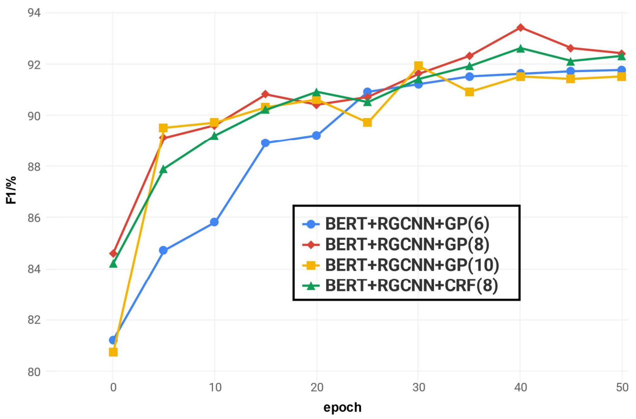

Figure 6. *Model training process.*

As shown in Figure 6, the training process illustrates the F1 values of the RGCNN at different depths with the number of rounds, where the first three groups are the training results of the Bert+RGCNN+GP model with depths of 6, 8, and 10 layers. The F1 value of the 10 layers model reached a maximum of 92.57% at 30 epochs.

It's found that in comparison between groups 2 and 4, the decoding layer using GP gives better results than CRF because the loss function and evaluation metrics of GP are entity-based, which works well on the entity-level dataset of MSRA, but the improvement is not obvious on tag-level data. The F1 value of BERT+RGCNN+GP is 4.3% higher than that of the BiLSTM+CRF model which is the baseline.

4.5. Comparison with Existing Work

Table 4. *Mainstream Model Comparison.*

Models	P	R	F1	Single step/s
Bert-finetuning	94.09	94.54	94.31	1363
BERT-BiLSTM-CRF	93.18	93.96	93.11	536
BERT-IDCNN-CRF	94.86	93.97	94.42	216
BERT-softlexion-RGCNN -GP	95.59	94.35	94.96	348
Latice-LSTM-CRF	93.57	92.79	93.18	7506
Radical-BiLSTM-CRF	91.28	90.62	90.96	>410

Models in Table 1 make comparisons with ones after adding the Bert layer in Table 4, which shows that a priori semantics of the pre-trained models significantly improves the evaluation metrics of all the base models. Moreover, among all the models using BERT in Table 4, BERT-IDCNN-CRF has the least single-step elapsed time and BERT-RGCNN-CRF is closest to its time. Our model's rapid single-step time is attributed to lessening training parameters. The number of parameters of BERT pre-trained language model is more than 100 million, and BERT-finetuning updates all parameters, while the RGCNN model combined with BERT fixing parameters of the BERT layer, only updates the upper layer parameters. Therefore, the number of parameters of RGCNN is 59,000 is significantly reduced to 59,000.

5. Conclusion

To address the problem of easy disappearance of gradients between layers of extracted entities and insufficient access to contextual information by CNN models, residual gated connections and attention mechanisms are appended on the basis of one-dimensional expanded convolution in order to obtain contextual semantic information while maintaining the training speed of CNN architecture. The BERT-RGCNN-GP model achieves an F1 value of 94.96% for entity extraction on MSRA. The feasibility of the experiment and the method is verified, hence next step is to consider adding the identification between semantic relations to further enhance the entity extraction effect.

References

[1] Jianlin Su. (May. 01, 2021). GlobalPointer: A unified approach to nested and non-nested NERs [Blog post]. Retrieved from https://spaces.ac.cn/archives/8373

[2] LI L S, HE H L, LIU S, et al. Biomedical named entity recognition based on word representation method [J]. Journal of Chinese Computer Systems, 2016, 37 (2): 302-307.

[3] WANG J, LI Y, JIANG X C, et al. Named entity recognition of LSTM based on hierarchical residual connection [J]. Journal of Jiangsu University (Natural Science Edition), 2022, 43 (4): 446-452.

[4] XU X B, WANG T, KANG R, et al. Multi-feature Chinese named entity recognition [J]. Journal of Sichuan University (Natural Science Edition), 2022, 59 (2): 022003.

[5] WANG H B, GAO H K, SHEN Q, et al. Thai language names, place names, and organization names entity recognition [J]. Journal of System Simulation, 2019, 31 (5). 1010-1018.

[6] LI N. Automatic extraction of alias in ancient local chronicles based on conditional random fields [J]. Journal of Chinese Information Processing, 2018, 32 (11): 41-48.

[7] ZOU B W, QIAN Z, CHEN Z C, et al. Negation and un-certainty information extraction oriented to natural language text [J]. Journal of Software, 2016, 27 (2): 309-328.

[8] Collobert R, Weston J, Bottou L, et al. Natural Language Processing (Almost) from Scratch [J]. The Journal of Machine Learning Research, 2011, 12 (1): 2493-253.

[9] HAMMERTON J. Named entity recognition with long short-term memory [C] //Conference on Natural Language Learning at HLT-NAACL. NJ. Association for Computational Linguistics, 2003.

[10] Huang Zhiheng, Xu Wei, Yu Kai. Bidirectional LSTM-CRF models for sequence tagging [J]. ArXiv Preprint ArXiv: 1508.01991, 2015: 1-10.

[11] LAMPLE G, BALLESTEROS M, SUBRAMANIAN S, et al. Neural architectures for named entity recognition [J/OL]. arXiv: 1603. 01360 [cs]. 2016.

[12] MA X, HOVY E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF [J/OL]. arXiv: 1603. 01354 [cs]. 2016.

[13] CHIU J P C, NICHOLS E. Named entity recognition with bidirectional LSTM-CNNs [J]. Transactions of the Association for Computational Linguistics, 2016 (4): 357-370.

[14] DONG C H, ZHANG J J, ZONG C Q, et al. Character-based LSTM-CRF with radical-level features for Chinese named entity recognition [M]// Natural Language Understanding and Intelligent Applications. Cham: Springer, 2016: 239-250.

[15] Peng M, Ma R, Zhang Q, et al. Simplify the Usage of Lexicon in Chinese NER [J]. ArXiv: 1908.05969v1, 2019.

[16] ZHANG Y, YANG J. Chinese NER using lattice LSTM [J /OL]. arXiv: 1805. 02023 [cs], 2018.

[17] Yang F, Zhang J, Liu G, et al. Five-Stroke Based CNN-BiRNN-CRF Network for Chinese Named Entity Recognition [M]. Hohhot, China: 7th CCF International Conference, 2018.

[18] STRUBELL E, VERGA P, BELANGER D, et al. Fast and accurate entity recognition with iterated dilated convolutions [J/OL]. arXiv: 1702.02098 [cs], 2017.

[19] Chen H, Lin Z, Ding G, et al. GRN: Gated relation network to enhance convolutional neural network for named entity recognition //Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33 (1): 6236-6243.

[20] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv: 1810.04805, 2018.