New Research on Transfer Learning Model of Named Entity Recognition

Guoliang Guan and Min Zhu*
School of Computer Science and Software Engineering, East China Normal University, Shanghai, 200062, China

*E-mail: mzhu@cc.ecnu.edu.cn

Abstract. This paper integrates the current Google's most powerful NLP transfer learning model BERT with the traditional state-of-the-art BiLSTM-CRF model to solve the problem of named entity recognition. A bi-directional LSTM model can consider an effectively infinite amount of context on both sides of a word and eliminates the problem of limited context that applies to any feed-forward models. Google’s model applied a feedforward neural network, causing its performance to weaken. We seek to solve these issues by proposing a more powerful neural network model named BT-BiLSTM. The new neural network model has obtained F1 scores on three Chinese datasets exceeds the previous BiLSTM-CRF model, especially on the value of recall. It shows the great value of the combination of large scale none-labelled data pre-trained language model with named entity recognition, which inspire new ideas on other future work.

1. Introduction
NER is also known as a proper name recognition. It generally refers to an entity with a specific meaning or strong reference in the text. Academically, NER usually includes persons’ names, locations or organization names. [1] The NER system extracts the above entities from the unstructured input text, and can identify more categories of entities according to needs. BERT applied a feedforward neural network, causing its performance to weaken. We seek to solve these issues by proposing a more powerful neural network model named BT-BiLSTM. A bi-directional LSTM model can consider an effectively infinite amount of context on both sides of a word and eliminates the problem of limited context that applies to any feed-forward models [2]. This article mainly focuses on chinese NER using CWS (chinese word segmentation) for sequential labelling [3]. In previous papers, named entity recognition only achieved good results in limited text types (mainly in news corpus) and entity categories (mainly names of people, places, and organizations) [4]. Compared with other information retrieval fields, entity naming evaluation is expected to be small and easy to produce over-fitting. The original BiLSTM model pays more attention to high accuracy, but the recall rate is not ideal, resulting in poor system performance for universal recognition of multiple types of named entities. In this paper, we adopted a new approach to improve the F1-score and Recall on NER.

2. Related work

2.1. Context Encoder Architectures

2.1.1. Pre-training. In the previous pre-training models (including word2vec, ELMo, etc.), word vectors
are generated. From input sequence to predicted tags, a DL-based NER model consists of distributed representations for input, context encoder, and tag decoder, illustrated in figure1. But if a large number of parameters are pre-trained through a large training set such as BERT, the majority of the network structure parameters are initialized, and then the poor data volume is compared with the Fine-tuning process to adjust the parameters to make them more suitable for solving new tasks, which makes research work become simpler. In recent years, the common pre-training models include ULMFiT, GPT and BERT, etc. The taxonomy of DL-based NER

2.1.2. Transformer. Transformer was proposed by Vaswani et al. [5], dispenses with recurrence and convolutions entirely. Transformer utilizes stacked self-attention and point-wise, fully connected layers to build basic blocks for encoder and decoder. Experiments on various tasks [5][6][7] show Transformer to be superior in quality while requiring significantly less time to train. Based on transformer, Radford et al. proposed Generative Pre-trained Transformer (GPT) [8] for language understanding tasks. GPT has a two-stage training procedure. First, they use a language model with Transformers on unlabeled data to learn the initial parameters. Then they adapt these parameters to a target task using the supervised objective, resulting in minimal changes to the pre-trained model. Peters et al. proposed ELMo [9] representations, which are computed on top of two-layer bidirectional language models with character convolutions. This new type of deep contextualized word representation is capable of modelling both complex characteristics of word usage (e.g., semantics and syntax), and usage variations across linguistic contexts (e.g., polysemy). Unlike GPT (a left-to-right architecture), Bidirectional Encoder Representations from Transformers (BERT) [10] is proposed to pre-train deep bidirectional Transformer by jointly conditioning on both left and right context in all layers. The pre-trained BERT representations can be fine-tuned with one additional output layer for a wide range of tasks including NER and chunking.

![Figure 1. The taxonomy of DL-based Chinese NER.](image)

2.2. Tag Decoder Architectures

2.2.1. Multi-Layer Perceptron. NER is in general formulated as a sequence labelling problem. With a multi-layer Perceptron +SoftMax layer as the tag decoder layer, the sequence labelling task is cast as a multi-class classification problem. Tagging for each word is independently predicted based on the context-dependent representations without considering its neighbors. A number of NER models, that has been introduced earlier use MLP + SoftMax as the tag decoder.

2.2.2. Conditional Random Fields. Based on these features, many machine learning algorithms have been applied in supervised NER, including Hidden Markov Models (HMM) [11], Decision Trees [12], Maximum Entropy Models [13], Support Vector Machines (SVM) [14], and Conditional Random Fields (CRF) [15]. A conditional random field (CRF) is a random field globally conditioned on the observation
sequence. CRFs have been widely used in feature-based supervised learning approaches. Many deep learning based NER models use a CRF layer as the tag decoder, e.g., on top of a bidirectional LSTM layer [16], and on top of a CNN layer [17]. Listed in Table 1, CRF is the most common choice for tag decoder, and the state-of-the-art performance is achieved by [18] with a CRF tag decoder. The work by Huang et al. [19] is among the first to utilize a bidirectional LSTM CRF architecture to sequence tagging tasks (POS, chunking and NER).

The reasons using CRF, rather than HMM model, are given below.

- The HMM is assumed to satisfy the HMM independent assumption. CRF does not, so CRF can accommodate more contextual information.
- The CRF calculates the global optimal solution, not the local optimal value.
- CRF is the joint probability of calculating the entire marker sequence given the observed sequence. The HMM is given the current state and the next state is calculated.
- CRF compares the selection of features and the format of feature functions, and the amount of training is large.

Table 1. Summary of recent related works on neural NER

| Model          | Input representation | Context encoder | Tag decoder | Performance (F-score) |
|----------------|----------------------|-----------------|-------------|------------------------|
| ELMo [9]       | CNN-LSTM-LM          | Word            | Hybrid      | LSTM CRF               | 92.2        |
| BERT [10]      | -                    | Word Piece Segment, position | Transformer MLP | 92.8        |
| BiLSTM[20]     | CNN                  | SENNA Capitalization, lexicons | LSTM CRF | 91.6        |
| SS-multitask[21] | LSTM                | Word2vec - | LSTM CRF | 86.2        |
| ID-CNNs[22]    | -                    | SENNA Word shape | ID-CNN CRF | 90.7        |
| SLSM-GRSM[23]  | -                    | SENNA POS, gazetteers | CNN Semi-CRF | 90.9        |
| Mc-BiLSTM[24]  | LSTM                 | GloVe Syntactic | LSTM CRF | 91.1        |
| BT-BiLSTM      | LSTM                 | Word Piece Segment, position | Transformer CRF | 93.6        |

3. Model

Traditional approaches to NER are broadly classified into three main streams: rule-based, unsupervised learning, and feature-based supervised learning approaches. As to the techniques applied in NER, it briefly introduces the previous work. Deep-learning based approaches, which automatically discover representations needed for the classification or detection from raw input in an end-to-end manner. Brief descriptions are given below.

3.1. BiLSTM-CRF

In the BiLSTM-CRF model, four types of features were used for the NER task: spelling features, context features, word embeddings, and gazetteer features. Their experimental results showed that the extra features (i.e., gazetteers) boost tagging accuracy. The BiLSTM-CRF model by Chiu and Nichols [20] incorporated a bidirectional LSTM and a character-level CNN. Besides word embeddings, the model used additional word-level features (capitalization, lexicons) and character-level features (4-dimensional vector representing the type of a character: upper case, lower case, punctuation, other).

The basic recurrent LSTM functions are:

\[
\begin{align*}
&\begin{bmatrix}
    f^c_j \\
    o^c_j \\
    i^c_j \\
    c^c_j \\
\end{bmatrix} =
\begin{bmatrix}
    \sigma \\
    \sigma \\
    \sigma \\
    \tanh
\end{bmatrix}
\begin{bmatrix}
    x^c_j \\
    h^c_{j-1}
\end{bmatrix} + b^c
\end{align*}
\]

(1)

\[
c^c_j = f^c_j \odot c^c_{j-1} + i^c_j \odot c^c_j
\]

(2)

\[
h^c_j = o^c_j \tanh \odot (c^c_j)
\]

(3)
Where \(i^f_j, f^f_j\) and \(o^f\) denote a set of input, forget and output gates, respectively. \(W^{cT}\) and \(b^c\) are model parameters. \(\sigma()\) represents the sigmoid function.

\[
\text{score}(l|s) = \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_j f_j(s, i, l, l_{i-1})
\]

The sentence \(s\) is the sentence we want to label the part of speech. \(i\) is used to represent the \(i^{th}\) word in sentence \(s\). \(l_j\) indicates the part of speech to be scored for the \(j^{th}\) word. \(l_{i-1}\) indicating the part of speech to be scored for the \((i-1)^{th}\) word. For RNN-based models, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two typical choices of the basic units. The LSTM network is generally called LSTM[25]. At present, a new combination named bi-directional long-short term memory (BiLSTM) network is a well-recognized network of better models.

### 3.2. BERT

The full name of BERT is Bidirectional Encoder Representations from Transformers, which is the encoder of the two-way Transformer, because the decoder can't get the information to be predicted. In fact, many of the design decisions in BERT were intentionally chosen to be as close to GPT as possible so that the two methods could be minimally compared. The main innovation of the model is the pre-train method, which uses Masked LM and Next Sentence Prediction to capture the word and sentence level representation respectively. In traditional pre-training, in order for a machine learning model to process words, it is first necessary to represent the words in a numerical form so that the model can be calculated, such as Word2vec word embedding. The first phase of Pre-training, similar to Word Embedding, trains a language model using existing unmarked corpus. The second phase is called fine-tuning, which uses a pre-trained language model to complete specific NLP downstream tasks. Data tag costs are always expensive because of the need for domain expertise and the process is time consuming, especially in NLP, where text understanding varies from person to person. However, there is a large amount (almost unlimited amount) of unlabeled data around us, and it can be easily extracted in order to greatly reduce the fine tune training time (the BERT is trained for 4 days on 4 TPUs). Because of this, the pre-trained network parameters, especially the underlying network parameters, are extracted from the specific task. Due to the reusability of the underlying features, the more the underlying is available, the NLP will be available regardless of the domain, and the more extracted features will be related to the task at hand. This is why we initialize the new task network parameters with the underlying pre-trained parameters. The high-level features are associated with the task. This paper uses Fine-tuning to clean up the high-level irrelevant feature extractor of BERT with the new data set.

### 3.3. BT-BiLSTM

This model uses the BERT model trained by Google to initialize the parameters of the model, so that the model can convergence faster in the training process. The original network layer needs to delete or add a new connection layer relationship. If all the parameters of the original network are directly changed, and then used as the pre-training model of this article, the parameter explosion phenomenon will occur, and the new training result is very poor. In this paper, the connection layer is modified and the original model is used for training. The model structure can be seen in figure2.

The method and attention are as follows:

- Delete the layer, and replace the feedforward neural network.
- The parameters of the layer that need to be left are not changed.
- Help to add new layer relationships, adjust the classifier network to have more output neurons, and then use SoftMax.
- Ensure that the data dimension relationships between the layers that are connected to each other are matched. For example, the data dimension of the original A input to the fully connected layer is 768*7*7, and the dimension input to the FC layer after modifying the network is also 768*7*7, which is consistent.

The data dimension of the output can be changed using the fully connected layer.

Previously, the BiLSTM model used word2vec. This paper mainly replaces the embedding layer of the original paper with BERT.
The BT-BiLSTM is used for tagging named entities. Our Model extracts a fixed length feature vector from tokens. For each word, these vectors are concatenated and fed to the BiLSTM network and then to the output layers. The output layers decode output into a score for each tag category. The extracted features of each word are fed into a forward LSTM network and a backward LSTM network. The output of each network at each time step is decoded by a linear layer and a log-SoftMax layer into log-probabilities for each tag category. These two vectors are then simply added together to produce the final output.

4. Experiment

4.1. Experiment environment and parameter

4.1.1. Batch size, learning-rate. Choosing an appropriate batch size and learning rate make the model both random and easier to converge, which makes the training the fastest convergence or the best convergence (global best) and reduces the risk of local optimization.

4.1.2. Dropout. Dropout randomly removes some of the cells of the hidden layer in order to prevent overfitting. In the training, for the neural network unit, according to a certain probability (if dropout=0.5, that is, removed with 50% probability, stop working), it is temporarily discarded from the network. In order to cover all the input information, more convolutional layers need to be added, resulting in deeper and deeper layers and more and more parameters. In order to prevent over-fitting, more regularizations such as Dropout should be taken into consideration. The hyperparameters were showed in Table 2.

| Hyperparameters          | Value   |
|--------------------------|---------|
| learning-rate            | 1E-5    |
| Batch size               | 64      |
| Dropout rate             | 0.5     |
| Warm up protection       | 0.1     |

4.2. Datasets

| Dataset Name               | Train  | dev  | test  |
|----------------------------|--------|------|-------|
| People’s Daily classic data| 1400K  | 1114K| 15K   |
| schema06                   | 13579K | 1642K| 1089K |
| Private data               | 6277K  | 702K | 1405K |
We use BIOES tagging scheme for Chinese NER tagging [26]. The experiment datasets are showed in Table 3. All the Words in the sentence are obtained using a Chinese segment. Use BIOES data annotation mode to use People's Daily classic data. To make this compatible with Word Piece tokenization, we feed each date tokenized input word into our Word Piece tokenizer and use the hidden state corresponding to the first sub-token as input to the classifier. The popular labelling scheme is ‘BIOES’: ‘B’ for the beginning character of a word, ‘I’ for the internal characters, ‘E’ for the ending character, and ‘S’ for single-character word, '-' or ‘O’ means that the metric does not apply [27]. Example appears in Figure 3:

| Text   | Hayao Tada , commander of the Japanese North China Area Army |
|--------|-------------------------------------------------------------|
| LOC    | - - - - B I - S - - -                                      |
| MISC   | - - - S B B I S S S S                                      |
| ORG    | - - - - - B I E I I E                                      |
| PERS   | B E - - - - - - S - -                                      |

*Figure 3. Example of how lexicon features are applied.*

4.3. Experiment result

NER involves identifying both entity boundaries and entity types. With “exact-match evaluation”, a named entity is considered correctly recognized only if it’s both boundaries and type match ground truth. Precision, Recall, and F-score are computed on the number of true positives (TP), false positives (FP), and false negatives (FN).

- True Positive (TP): entities that are recognized by NER and match ground truth.
- False Positive (FP): entities that are recognized by NER but do not match ground truth.
- False Negative (FN): entities annotated in the ground truth that are not recognized by NER.

Precision measures the ability of a NER system to present only correct entities, and Recall measures the ability of a NER system to recognize all entities in a corpus. F1-score is the harmonic mean of precision and recall, and the balanced F1-score is most commonly used:

\[
Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}
\]

*Figure 4. Loss, precision, recall within step change on dev data by BT-BiLSTM using tensor board.*

Loss, precision, recall graphs can be seen in Figure 4. Test-F1 is the NER F-measure in the testing datasets. BT-BiLSTM model performed well on F1 and recall value on below three Chinese datasets. The Dev and Test scores are averaged over 5 random restarts using those hyperparameters. Results are presented in Table 4 and Table 5.

| Table 4. Results on datasets. |
|-----------------------------|
| BiLSTM-CRF | BERT-base | BT-BiLSTM |
| daily | Schema06 | daily | Schema06 | daily | Schema06 |
| Accuracy | 98.7 | 98.5 | 99.0 | 88.7 | 99.7 | 99.3 |
| Precision | 92.1 | 90.6 | 92.4 | 91.5 | 92.2 | 92.5 |
| Recall | 89.8 | 88.7 | 87.5 | 87.1 | 91.0 | 94.7 |
| Test F1 | 91.0 | 89.7 | 90.0 | 89.3 | 91.6 | 93.6 |
Table 5. NER(CWS) Results in F-measure.

|        | BiLSTM-CRF |          |          | BT-BiLSTM |          |          |
|--------|------------|----------|----------|-----------|----------|----------|
| LOC    |            |          |          |           |          |          |
| Private| 92.0       | 91.7     | 91.9     | 91.0      | 95.0     | 93.0     |
| Daily  | 93.8       | 90.8     | 92.3     | 94.2      | 94.0     | 94.1     |
| Schema06| 93.5       | 88.0     | 90.7     | 94.3      | 94.4     | 94.4     |
| ORG    |            |          |          |           |          |          |
| Private| 89.1       | 83.9     | 86.5     | 81.8      | 89.8     | 85.7     |
| Daily  | 87.6       | 87.7     | 87.6     | 88.9      | 91.8     | 90.4     |
| Schema06| 83.0       | 88.7     | 85.7     | 84.1      | 92.8     | 88.2     |
| PER    |            |          |          |           |          |          |
| Private| 93.8       | 93.6     | 93.7     | 96.0      | 96.8     | 96.4     |
| Daily  | 92.8       | 89.8     | 91.3     | 94.1      | 90.0     | 92.1     |
| Schema06| 92.1       | 89.8     | 91.0     | 96.1      | 96.4     | 96.3     |
| Average| 90.9       | 89.3     | 90.1     | 91.2      | 94.4     | 92.7     |

a Location.  
b Organization.  
c Person.

5. Conclusion

According to the experiment, our new model BT-BiLSTM has achieved nice scores on F1-score and Recall, showing a new research direction. Grammar-based systems usually get better accuracy, but at the cost of low recalls. In contrast, this model improves the recall rate very well. What should be pointed is the operation speed and the time costs for the optimization. DL-based NER models achieved good performance with the cost of massive computing power. For example, ELMo representation represents each word with a $3 \times 1024$-dimensional vector, and the model was trained for 5 weeks on 32 GPUs. Google BERT representations were trained on 64 cloud TPUs. The training curve of the model shows that the convergence is very slow, and there is room for further optimization. Developing approaches to balancing model complexity and scalability will be a promising direction. On the other hand, model compression and pruning techniques are also options to reduce the space and computation time.

Acknowledgments

The authors would like to thank Kang He, Yu Liu for helpful discussions, and the anonymous reviewers for insightful comments.

References

[1] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Lingvisticae Investigations, vol. 30, no. 1*, pp.3–26, 2007.
[2] Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649.
[3] Wenliang Chen, Yujie Zhang, and Hitoshi Isahara.2006. Chinese named entity recognition with conditional random fields. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. pages 118–121.
[4] Z. Huang, W. Xu, and K. Yu, “Bidirectional lstm-crf models for sequence tagging,” *arXiv preprint arXiv:1508.01991*, 2015.
[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N.Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. NIPS*, 2017, pp. 5998–6008.
[6] N. Kitaev and D. Klein, “Constituency parsing with a selfattentive encoder,” in *Proc. ACL*, 2018, pp. 2675–2685.
[7] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, “Generating wikipedia by summarizing long sequences,” *arXiv preprint arXiv:1801.10198*, 2018.
[8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding
8

by generative pre-training,” Technical report, OpenAI, 2018.

[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proc. NAACL-HLT, 2018, pp. 2227–2237.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pretraining of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805, 2018.

[11] S. R. Eddy, “Hidden markov models,” Current opinion in structural biology, vol. 6, no. 3, pp. 361–365, 1996.

[12] J. R. Quinlan, “Induction of decision trees,” Machine learning, vol. 1, no. 1, pp. 81–106, 1986.

[13] J. N. Kapur, Maximum-entropy models in science and engineering. John Wiley & Sons, 1989.

[14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support vector machines,” IEEE Intelligent Systems and their applications, vol. 13, no. 4, pp. 18–28, 1998.

[15] J. Lafferty, A. McCallum, and F. C. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” 2001.

[16] Z. Huang, W. Xu, and K. Yu, “Bidirectional lstm-crf models for sequence tagging,” arXiv preprint arXiv:1508.01991, 2015.

[17] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” Journal of Machine Learning Research, vol. 12, no. Aug, pp. 2493–2537, 2011.

[18] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labeling,” in Proc. COLING, 2018, pp. 1638–1649.

[19] A. Akbik, D. Blythe, and R. Vollgraf, “Contextual string embeddings for sequence labeling,” in Proc. COLING, 2018, pp. 2145–2158.

[20] J. Zhuo, Y. Cao, J. Zhu, B. Zhang, and Z. Nie, “Segment-level sequence modeling using gated recursive semi-markov conditional random fields,” in Proc. ACL, vol. 1, 2016, pp. 1413–1423.

[21] E. Strubell, P. Verga, D. Belanger, and A. McCallum, “Fast and accurate entity recognition with iterated dilated convolutions,” Proc. ACL, 2017, pp. 2670–2680.

[22] J. Zhuo, Y. Cao, J. Zhu, B. Zhang, and Z. Nie, “Segment-level sequence modeling using gated recursive semi-markov conditional random fields,” in Proc. ACL, vol. 1, 2016, pp. 1413–1423.

[23] B. Y. Lin, F. Xu, Z. Luo, and K. Zhu, “Multi-channel bilstm-crf model for emerging named entity recognition in social media,” in Proc. the 3rd Workshop on Noisy User-generated Text, 2017, pp. 160–165.

[24] O. Kuru, O. A. Can, and D. Yuret, “Charner: Character-level named entity recognition,” in Proc. COLING, 2016, pp. 911–921.

[25] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, “Semi supervised sequence tagging with bidirectional language models” in Proc. ACL, 2017, pp. 1756–1765.

[26] Fan Yang and Paul Vozila. 2014. Semi-supervised Chinese word segmentation using partial-label learning with conditional random fields. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 90–98, Doha, Qatar, October. Association for Computational Linguistics.