Clinical Name Entity Recognition using Conditional Random Field with Augmented Features

Dawei Geng
(Intern at Philips Research China, Shanghai)

Abstract. In this paper, We presents a Chinese medical term recognition system submitted to the competition held by China Conference on Knowledge Graph and Semantic Computing. I compare the performance of Linear Chain Conditional Random Field (CRF) with that of Bi-Directional Long Short Term Memory (LSTM) with Convolutional Neural Network (CNN) and CRF layers performance and find that CRF with augmented features performs best with F1 0.927 on the offline competition dataset using cross-validation. Hence, this system was built by using a conditional random field model with linguistic features such as character identity, N-gram, and external dictionary features.

Keywords: Linear-Chain Conditional Random Field, Name Entity Recognition, Long Short Term Memory, Convolutional Neural Network

1 Introduction

Sequence tagging including part of speech tagging (POS), chunking, and named entity recognition (NER) has been a typical NLP task, which has drawn research attention for a few decades.

This task focuses on recognizing 5 categories name entities in the Chinese Clinical Notes provided by JiMuYun Health Technology company from Beijing, China. This task was part of the China Conference on Knowledge Graph and Semantic Computing conference. Data are real electronic clinical notes, consisting of 1198 notes with labeled name entities and 10003 unlabeled notes.

The remainder of the paper is organized as follows. The next section is a review of related work. A simple introduction to Conditional Random Field and Bi-directional LSTM with CNN and CRF layers is given in Section 3. In Section 4, experimental results are demonstrated. Conclusions are summarized in Section 5.

2 Related Work

Current methods for Clinical NER fall into four general classes, i.e., dictionary-based methods, heuristic rule-based methods, and statistical machine learning methods, and deep learning methods.

Relying on dictionary-based methods can cause the low recall due to the continual appearance of new entities with the advancing medical research. Clinical

named entities do not follow any nomenclature, which makes rule-based methods hard to be perfect. Besides, rule-based systems require domain experts, and they are not flexible to other NE types and domains.

Machine learning methods are more robust and they can identify potential biomedical entities which are not previously included in standard dictionaries. More and more machine learning methods are explored to solve the Bio-NER problem, such as Hidden Markov Model (HMM), Support Vector Machine (SVM), Maximum Entropy Markov Model (MEMM), and Conditional Random Fields (CRF) (Lafferty et al., 2001).

Further, many deep learning methods are employed to tag sequence data. For example, Convolutional network based models (Collobert et al., 2011) have been proposed to tackle sequence tagging problem. Such model consists of a convolutional network and a CRF layer on the output. In speech language understanding community, recurrent neural network (Mesnil et al., 2013; Yao et al., 2014) and convolutional nets (Xu and Sarikaya, 2013) based models have been recently proposed. Other relevant works include (Graves et al., 2005; Graves et al., 2013) which proposed a bidirectional recurrent neural network for speech recognition.

In this paper, I make a comparison between classical statistical machine learning method- Conditional Random Fields and deep learning method – Bi-LSTM with CNN and CRF layers in terms of their performance on the competition dataset.

3 Methodology

We make comparison between the performance of Conditional Random Field with designed linguistic features and that of Bi-directional LSTM with CNN and CRF layer on the competition dataset. CRF experiment is carried out using Python sklearn-crfsuite 0.3 package, LSTM is conducted using Tensorflow r1.2.

3.1 Labeling

In order to conduct supervised learning, we label the Chinese character sequence, There are 5 kinds of entities, which are encoded into 0 to 4. Since we label data on character level and entities usually consist of multiple characters, we label the beginning character of the entity B- with corresponding coded category, the rest of the entity character I- with corresponding coded category. If a character is not part of the entity, we label this character O. For example:”右肩左季肋部” is an entity belongs to Body Parts category. This entity is labeled as followed: 右 B-4, 肩I-4, 左I-4, 季I-4, 肋I-4, 部I-4

3.2 Conditional Random Field

Conditional Random Fields (CRFs) are undirected statistical graphical models, a special case of which is a linear chain that corresponds to a conditionally trained finite-state machine. Such models are well suited to sequence analysis, and CRFs in particular have been shown to be useful in part-of-speech tagging, shallow parsing, and named entity recognition for newswire data.

Let $o=\langle o_1, o_2, ...o_n \rangle$ be an sequence of observed words of length n. Let S be a set of states in a finite state machine, each corresponding to a label $l \in L$, Let $s = \langle s_1, s_2, ...s_n \rangle$ be the sequence of states in S that correspond to the labels assigned to words in the input sequence o. Linear chain CRFs define the conditional probability of a state sequence given an input sequence to be:

$$P(s \mid o) = \frac{1}{Z_0} \exp(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(s_{i-1}, s_i, o, i))$$

where Z_0 is a normalization factor of all state sequences, $f_j(s_{i-1}, s_i, o, i)$ is one of m functions that describes a feature, and λ_j is a learned weight for each such feature function. This paper considers the case of CRFs that use a first order Markov independence assumption with binary feature functions.

Intuitively, the learned feature weight λ_j for each feature f_j should be positive for features that are correlated with the target label, negative for features that are anti-correlated with the label, and near zero for relatively uninformative features. These weights are set to maximize the conditional log likelihood of labeled sequences in a training set $D = \{\langle o, l \rangle_{(1)}, ..., \langle o, l \rangle_{(n)}\}$:

$$LL(D) = \sum_{i=1}^{n} \log(P(l_{(i)} \mid o_{(i)})) - \sum_{j=1}^{m} \frac{\lambda_j^2}{2\sigma^2}$$

When the training state sequences are fully labeled and unambiguous, objective function is convex, thus the model is guaranteed to find the optimal weight settings in terms of LL(D). Once these settings are found, the labeling for a new, unlabeled sequence can be done using a modified Viterbi algorithm. CRFs are presented in more complete detail by Lafferty et al. (2001).

3.2 Bi-directional LSTM with CNN and CRF layer

3.2.1 Convolutional Neural Network layer

Convolution is widely used in sentence modeling to extract features. Generally, let l and d be the length of sentence and word vector, respectively. Let $C \in \mathbb{R}^{d \times l}$ be the sentence matrix. A convolution operation involves a convolutional

kernel $H \in \mathbb{R}^{d \times w}$ which is applied to a window of w words to produce a new feature. For instance, a feature ci is generated from a window of words $C[*,\ i:i+w]$ by

$$c_i = \sigma(\sum(C[*,\ i:i+w] \circ H) + b)$$

Here $b \in \mathbb{R}$ is a bias term and σ is a non-linear function, normally tanh or ReLu. $\circ$ is the Hadamard product between two matrices. The convolutional kernel is applied to each possible window of words in the sentence to produce a feature map. $c = [c_1, c_2, ..., c_{l-w+1}]$ with $c \in \mathbb{R}^{l-w+1}$.

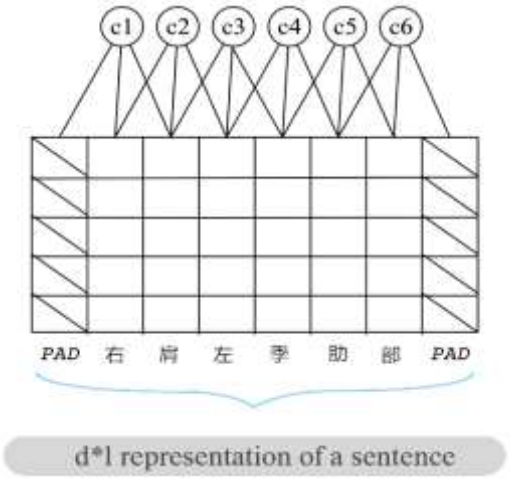

Figure 3 Convolution Neural Network

Next, I apply pairwise max pooling operation over the feature map to capture the most important feature. The pooling operation can be considered as feature selection in natural language processing.

Specifically, the output of convolution, the feature map $c = [c_1, c_2, ..., c_{l-w+1}]$ is the input of the pooling operation. The adjacent two features in the feature map be calculated as follows:

$$p_i = \max(c_{i-1}, c_i)$$

The output of the max pooling operation is $p = [p_1, p_2, ..., p_l],\ p \in \mathbb{R}^l$ p_i captures the neighborhood information around character i within a window of specified step size. If I apply 100 different kernels, 2 different step sizes to extract features, then p will become $\mathbb{R}^{200*l}$. Then I concatenate convolutional features to their corresponding original character features (word2vec features) to get feature sentence matrix.

3.2.2 Bi-directional Long Short Term Memory

Long Short Term Memory is a special kind of Recurrent Neural Network. It can maintain a memory based on history information using purpose-built memory cells,

which enables the model to predict the current output conditioned on long distance features. LSTM memory cell is implemented as the following:

$$i_t = \sigma(W_{xi}x_i + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$
$$h_t = o_t \tanh(c_t)$$

Where sigma is logistic sigmoid function and i, f, o and c are the input gate, forget gate, output gate, and cell vectors all of which are the same size as the hidden vector h. The weight matrix subscripts have the meaning as the name suggests. For example, Whi is the hidden-input gate matrix, Wxo is the input-output gate matrix etc. The weight matrices from the cell to gate vectors (e.g. Wci) are diagonal, so element m in each gate vector only receives input from element m of the cell vector. LSTM cell's structure is illustrated in Figure 1.

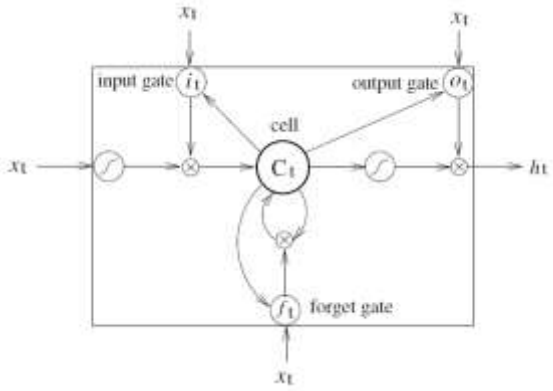

Fig. 1 - LSTM CELL

Here we use bi-directional LSTM as proposed in (Graves et al., 2013) because both past and future input features for a given time could be accessed. In doing so, we can efficiently make use of past features (via forward states) and future features (via backward states) for a specific time frame. (Below dashed boxes are the LSTM cells)

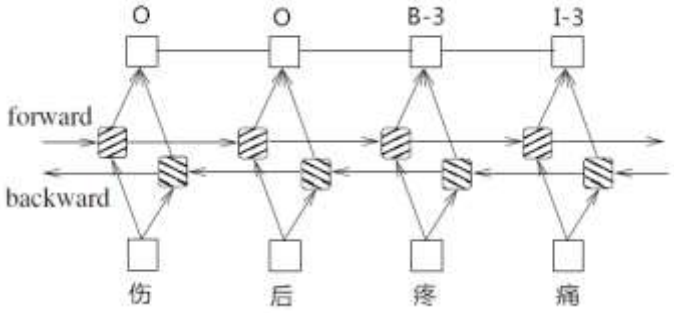

Fig. 2 - Bi-directional LSTM

4 Experiments

4.1 Features Template For CRF

For convenience, features are generally organized into some groups called feature templates. For example, a bigram feature template C1 stands for the next character occurring in the corpus after each character.

Table 1. Feature Template

Type	Feature	Function
Unigram	C-2,C-1,C0,C1,C2	The previous, current, and next character
Bigram	C-2C-1,C-1C0, C0C1,C1C2	The previous (next) and current characters
Trigram	C-2C-1C0,C-1C-0C1, C0C1C2	Possible 3 continuous chars in the feature window
Punctuation, Digits, Alphabets	IsAlpha, IsPunc, Isdigits	Current char is punctuation/digits/alpha or not
Position of Char	Bos, Eos	If char is in the start/end of the sentence
Common Suffix	From external dictionary	If character in common suffix
Common Prefix	From external dictionary	If character in common prefix

4.1.1 External Dictionary

We use dictionary such as ICD10, ICD9, and other Medicine, Pathology dictionary (72000 terms in total)to summarize common bigram, trigram, 4gram prefix and suffix. For example, if a bigram from sentence appears in common prefix or suffix, we make this feature 1 otherwise 0.

4.2 Hyperparameter Tuning

In Conditional Random Field, we use Elastic Nets as regularizing term and set optimization algorithm as LBFGS and maximum iteration as 500. After random search to tune the regularization coefficients C1, C2, I get the best C1,C2 as 0.089 and 0.004.

As for deep learning models, we set the parameters as below:

Table 2. Hyperparamters

Hyperparameter	Bi- LSTM +CRF	Bi- LSTM +CNN+CRF
n _features	1	1
max_length	1300	1300
#CNN_Kernel	None	100
step size	None	3,5
hidden_size	600	800
n_epochs	50	50
batch_size	100	100
n_classes	11	11
max_grad_norm	10	10
lr	0.001	0.001
dropout	0.3	0.3
embed_size	60	60

4.3 Performance Comparison

We use cross-validation to evaluate F1 performance across models, here list only one validation result for your reference.

Table 3. F1 Performance from different models

Models	Precision	Recall	F1
Conditional Random Field(only character features)	92.85%	91.96%…	92.40
Conditional Random Field(all template features)	93.10%	92.37%	92.73
Bi-directional LSTM+CRF layer	84.19%	90.87%	87.40
Bi-directional LSTM with CNN, CRF layers	90.31%	92.40%	91.35

Using the hyperparameters listed above, we could see with basic character features such as unigram, bigram, trigram, CRF is able to perform better than bi-directional LSTM with CRF layers. With other augmented features, CRF's performance could be improved further. Convolutional Neural Network layers could help bi-directional LSTM extract features better and improve its performance, but still not as good as CRF.

5 Conclusion

Conditional Random Field with augmented features performs better in my experiment compared to Bi-directional LSTM with CNN and CRF layers in terms of F1 performance. Hence I used CRF with augmented features model for the competition.

Future works will concentrate on hyperparameter tuning for deep learning models to get a better sense of how good the model is.

5 Special Thanks

We want to give special thanks to Dr. Liang Tao's guidance during this competition.

References

1. John D. Lafferty, Andrew McCallum, and Fernando C. N.Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
2. Settles, B., 2004, August. Biomedical named entity recognition using conditional random fields and rich feature sets. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (pp. 104-107). Association for Computational Linguistics.
3. A. Graves and J. Schmidhuber. 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures.Neural Networks.
4. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa.
5. 2011. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research(JMLR)
6. P. Xu and R. Sarikaya. 2013.Convolutional neural network based triangular CRF for joint intent detection and slot filling. Proceedings of ASRU.
7. G. Mesnil, X. He, L. Deng, and Y. Bengio. 2013. Investigation of recurrent neural-network architectures and learning methods for language understanding. Proceedings of INTERSPEECH.
8. A. Graves, A. Mohamed, and G. Hinton. 2013. Speech Recognition with Deep Recurrent Neural Networks. arxiv.
9. K. S. Yao, B. L. Peng, G. Zweig, D. Yu,X. L. Li, and F. Gao. 2014. Recurrent conditional random fields for language understanding. ICASSP.
10. Huang, Z., Xu, W. and Yu, K., 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.
11. Wang, X., Jiang, W., Luo, Z. 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: 26th International Conference on Computational Linguistics, COLING 2016, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016, pp. 2428–2437