Clinical Named Entity Recognition: ECUST in the CCKS-2017 Shared Task 2

Yuhang Xia and Qi Wang(✉)

East China University of Science and Technology, Shanghai, China 200237
153996626@qq.com, dsx4602@163.com

Abstract. Clinical named entity recognition aims to identify and classify clinical terms in electronic medical records, including diseases, symptoms, treatments, exams, and body parts. Challenges occur due to ambiguity in the boundary of Chinese words and the number limitation of annotated training data. In this paper, we propose a Bi-LSTM CRF model along with self-taught learning, active learning, and ensemble learning to recognize clinical named entities. The results achieved on CCKS-2017 Task 2 dataset with a F_1-Measure of 89.88% ranks among the top systems.

Keywords: clinical named entity recognition, Bi-LSTM CRF, self-taught learning, active learning, ensemble learning

1 Introduction

Electronic medical record systems have been widely used in China. Many tasks in clinical text mining rely on accurate clinical named entity recognition (NER), the identification of text spans mentioning a concept of a specific class, including disease, symptom, exam, treatment, and body part. Challenges occur due to ambiguity in the boundary of Chinese words and number limitation of annotated training data.

Traditionally, most of the effective NER approaches are based on machine learning techniques, such as Support Vector Machines (SVM) [1], Hidden Markov Models (HMM) [2], Conditional Random Fields (CRF) [3], Convolutional Neural Network (CNN) based models [4], and Recurrent Neural Network (RNN) based models [5]. For biomedical NER tasks, existing efforts include rule or dictionary based methods [6], supervised methods [7], and distant supervision methods [8].

In this paper, we regard the clinical NER as a sequence labeling problem, and use the Bi-LSTM CRF model which is similar to the one presented by Huang et al. [5] to address the problem. Different from Huang et al., we exploit character embedding rather than word embedding to deal with the ambiguity in the boundary of Chinese words. In addition, self-taught learning and active learning is introduced to enlarge the training set. Finally, ensemble learning is used to obtain the best recognition performance for all five types of clinical named entities. The results achieved on CCKS-2017 Task 2 dataset with a F_1-Measure of 89.88% ranks among the top systems.

2 Problem Formalism

The clinical named entity recognition task is defined as a sequence labeling problem in this paper. Given a text sequence $X =< x_1, ..., x_n >$, the goal is to label X with tag sequence $Y =< y_1, ..., y_n >$. We experiment with three different tagging formats for the recognition, including BIO (Begin, Inside, Outside), BIOS (Begin, Inside, Outside, Single), and BIEOS (Begin, Inside, End, Outside, Single). Examples of the three tagging formats can be found in Table 1.

Table 1. The tag sequences for "腹平坦，未见腹壁静脉曲张。" with three different tagging formats. The B-tag indicates the beginning of an entity. The I-tag indicates the inside of an entity. The E-tag indicates the end of an entity. The O-tag indicates the character is outside an entity. The S-tag indicates the single character is an entity. As for entity types, the b-tag indicates the entity is a body part, and the s-tag indicates the entity is a symptom.

	腹	平	坦	，	未	见	腹	壁	静	脉	曲	张	。
BIO	B-b	O	O	O	O	O	B-b	I-b	B-s	I-s	I-s	I-s	O
BIOS	S-b	O	O	O	O	O	B-b	I-b	B-s	I-s	I-s	I-s	O
BIEOS	S-b	O	O	O	O	O	B-b	E-b	B-s	I-s	I-s	E-s	O

3 Methods

In this section, we will describe the methods we employ, including Bi-LSTM CRF, self-taught learning, active learning, and ensemble learning.

3.1 Bi-LSTM CRF

This model is similar to the one presented by Huang et. al. [5]. It combines the framework of bidirectional LSTM layer [9] with linear chain CRF [10]. Different from Huang et. al., we employ character embedding rather than word embedding to deal with the ambiguity in the boundary of Chinese words.

The raw natural language input sentence is processed into sequence of characters $\mathrm{X} = [x]_1^T$. The character sequence is fed into an embedding layer, which produces dense vector representation of characters. The character vectors are then fed into a bidirectional LSTM layer. The LSTM [11] incorporates a gated memory-cell to capture long-range dependencies within the data. In the bidirectional LSTM, for any given sequence, the network computes both a left, $\overrightarrow{h_t}$, and a right, $\overleftarrow{h_t}$, representations of the sequence context at every input, x_t. The final representation is created by concatenating them as $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$. The bidirectional LSTM along with the embedding layer is the main machinery responsible for learning a good feature representation of the data.

Then the network use sentence level tag information via a CRF layer fed by a fully connected hidden layer. The CRF layer is represented by lines which connect

consecutive output layers, and has a state transition matrix as parameters. With such a layer, we can efficiently use past and future tags to predict the current tag, which is similar to the use of past and future input features via a bidirectional LSTM network. We consider the matrix of scores $f_\theta([x]_1^T)$ are output by the network. The element $[f_\theta]_{i,t}$ of the matrix is the score output by the network with parameters θ, for the sentence $[x]_1^T$ and for the i-th tag, at the t-th character. We introduce a transition score $[A]_{i,j}$ to model the transition from i-th state to j-th for a pair of consecutive time steps. Note that this transition matrix is position independent. We now denote the new parameters for our network as $\tilde{\theta} = \theta \cup \{[A]_{i,j} \forall i, j\}$. The score of a sentence $[x]_1^T$ along with a path of tags $[i]_1^T$ is then given by the sum of transition scores and network scores:

$$S([x]_1^T, [i]_1^T, \tilde{\theta}) = \sum_{t=1}^{T} ([A]_{[i]_{t-1},[i]_t} + [f_\theta]_{[i]_t,t}) \tag{1}$$

The dynamic programming [12] can be used efficiently to compute $[A]_{i,j}$ and optimal tag sequences for inference. See [10] for details.

3.2 Add Training Data by Self-taught Learning

Generally, the more training data we have, the better performance we can get. Considering that we have some unlabeled sentences, a self-taught learning algorithm is introduced to enlarge the training set. First, we train a Bi-LSTM CRF model using the original training set. Second, we apply the trained model to annotate the unlabeled sentences. Third, we choose some high-quality annotation results and add them to the original training set. Specifically, we define an annotating confidence (AC) to evaluate the quality of the annotation results:

$$AC([x]_1^T, [y]_1^T, \tilde{\theta}) = \frac{e^{S([x]_1^T, [y]_1^T, \tilde{\theta})}}{\sum_j e^{S([x]_1^T, [j]_1^T, \tilde{\theta})}} \tag{2}$$

where $[y]_1^T$ is the tag sequence predicted by the trained model and $[j]_1^T$ is the set of all possible output sequences. We choose those annotation results whose AC is greater than a threshold. In addition, considering that disease and treatment entities are much fewer than symptom and exam entities, and the body part entity recognition is not well performed, we only choose the annotation results which must include disease, treatment, or body part entities.

3.3 Add Training Data by Active Learning

We also use active learning algorithm to improve the recognition performance. First, we use the original training data along with the self-taught high-quality annotation results to train a Bi-LSTM CRF model. Second, we apply the trained model to annotate the rest unlabeled sentences. Third, we manually re-label a few low-quality annotation results whose AC is less than a threshold, and then add them to the training set. Similar to the self-taught algorithm, we only choose the annotation results which must include disease, treatment, or body part entities to re-label manually.

3.4 Improve Recognition Performance through Ensemble Learning

The task has five types of clinical named entities to be recognized. If we train models for each type of entities respectively, a class imbalance problem (too many O-tags in sequences) will exist, and the models cannot utilize the information of other types of entities. Thus, we train models to annotate all five types of entities at the same time, but finally choose five models in which each model has the best performance for the corresponding type on validation data, so that we can obtain the best recognition performance for all five types of entities. For example, for disease entities, we choose the model which has the best performance in recognizing diseases, and only use its disease recognition results as part of the final recognition results.

4 Experiments

4.1 Dataset and Evaluation Metrics

We use the CCKS-2017 Task 2 dataset to perform our experiments. The dataset contains 10,420 unannotated instances and 1,596 annotated instances with five types of clinical named entities, including diseases, symptoms, exams, treatments, and body parts. The annotated instances are already partitioned into 1,198 training instances and 398 test instances. Each instance has one or several sentences. We further partition the training sentences, take 70% of them as training data, and the rest 30% as validation data. We score our methods by using the CCKS-2017 Task 2 official metrics, which computes F_1-Measure.

4.2 Hyper-parameters and Training Details

As for the Bi-LSTM CRF model, we initialize character embeddings via word2vec[13] on both the annotated data and the unannotated data. Each character embedding is 100-dimensional. We compare the results with word embedding segmented by Jieba Chinese segmentation module, which is also 100-dimensional. We set the size of LSTM hidden layer to 64, and apply dropout [14] to the output of the Bi-LSTM layer. The dropout rate is 0.2. The Bi-LSTM CRF model is trained by AdaDelta [15] and the batch size is 128. For self-taught learning, we add the top 3,850 automatically annotated sentences to the training set. For active learning, we re-label the worst 125 automatically annotated sentences manually. We split training data into sentences by periods, and set the maximum length of a sentence to 185. If a sentence is longer than 185 characters, it will be further split into clauses. Then we take all the labeled sentences and clauses as a whole data set after de-duplication. We also try to split all the training data into clauses by periods, commas and semicolons to train different models.

4.3 Results and Discussion

To explore the impact of different settings, we train Bi-LSTM CRF models using different tagging formats on training data both in clauses and sentences, and

test them on validation data. The results are shown in Table 2. First, character-level models outperform word-level models. It is because word-level approaches may have segmentation error. What's more, the word set is much bigger than the character set. This means the corpus is not big enough to learn word embeddings effectively. Second, the recognition of diseases, treatments, and body parts is not well performed, which indicates the necessity of self-taught learning and active learning. Third, different types of entities are recognized the best by different settings, respectively. This indicates the necessity of ensemble learning.

Table 2. Performance (F_1-Measure) of Bi-LSTM CRF models with different settings tested on validation data. Note that we report the best results for each type, which means different lines are reported by different models.

	tag in character level						tag in word level					
	train in clauses			train in sentences			train in clauses			train in sentences		
	BIO	BIOS	BIEOS	BIO	BIOS	BIEOS	BIO	BIOS	BIEOS	BIO	BIOS	BIEOS
disease	77.54	75.00	77.21	74.15	75.20	**77.78**	64.08	**64.74**	64.08	63.44	62.96	63.84
symptom	96.24	96.21	**96.37**	96.26	96.17	96.27	94.02	94.64	**94.80**	93.27	93.99	93.75
exam	95.21	**95.57**	95.54	94.92	94.80	95.09	84.32	84.68	**84.93**	83.67	83.59	83.91
treatment	79.65	80.71	**82.37**	78.94	77.97	79.59	74.89	76.37	**77.10**	74.78	76.40	75.19
body part	83.85	**83.94**	83.19	82.67	83.39	83.81	72.59	73.41	**73.56**	71.87	72.57	73.07
overall	88.62	90.29	90.28	88.33	89.62	**90.33**	78.57	82.10	**82.47**	77.97	81.37	81.52

To dissect the effectiveness of self-taught learning, active learning, and ensemble learning, we train Bi-LSTM CRF models in character level only, and test them on test data. Table 3 shows that the Bi-LSTM CRF model along with self-taught learning, active learning, and ensemble learning achieves the best performance in clinical NER, with a F_1-Measure of 89.88%.

Table 3. Performance (F_1-Measure) of Bi-LSTM CRF models with different settings tested on test data.

	train in clauses			train in sentences		
	BIO	BIOS	BIEOS	BIO	BIOS	BIEOS
Bi-LSTM + CRF	87.81	88.36	**88.47**	87.09	87.74	87.87
+ self-taught learning (only)	88.44	88.49	**88.67**	87.75	87.60	88.12
+ active learning (only)	88.17	88.57	**88.79**	87.63	87.90	88.27
+ self-taught and active learning	88.72	88.63	**88.83**	87.78	88.35	88.53
+ ensemble learning	**89.88**					

5 Conclusion

In this paper, we propose a Bi-LSTM CRF model along with self-taught learning, active learning, and ensemble learning to recognize clinical named entities. We exploit character embedding to deal with the ambiguity in the boundary of Chinese words, and employ self-taught learning and active learning to increase

training data. After comparing different tagging schemes, we use ensemble learning to obtain the best recognition performance for all five types of entities. The results achieved on CCKS-2017 Task 2 dataset ranks among the top systems.

Acknowledgements. This work is supported by the 863 Program funded by China Ministry of Science and Technology (Program No.2015AA020107).

References

1. Asahara, M., Matsumoto, Y.: Japanese named entity extraction with redundant morphological analysis. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics (2003) 8–15
2. Bikel, D.M., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: Proceedings of the fifth conference on Applied natural language processing, Association for Computational Linguistics (1997) 194–201
3. Mccallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Conference on Natural Language Learning at Hlt-Naacl. (2003) 188–191
4. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research **12**(1) (2011) 2493–2537
5. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. Computer Science (2015)
6. Kipper-Schuler, K., Kaggal, V., Masanz, J., Ogren, P., Savova, G.: System evaluation on a named entity corpus from clinical notes. In: International Conference on Language Resources and Evaluation 2008. (2008) 3007–3011
7. Wang, Y., Yu, Z., Chen, L., Chen, Y., Liu, Y., Hu, X., Jiang, Y.: Supervised methods for symptom name recognition in free-text clinical records of traditional chinese medicine: An empirical study. Journal of biomedical informatics **47** (2014) 91–104
8. Bing, L., Ling, M., Wang, R.C., Cohen, W.W.: Distant ie by bootstrapping using lists and document structure. arXiv preprint arXiv:1601.00620 (2016)
9. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks the Official Journal of the International Neural Network Society **18**(5) (2005) 602–610
10. Lafferty, John, D., McCallum, Andrew, Pereira, Fernando, C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. (2001)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997) 1735–1780
12. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Readings in Speech Recognition **77**(2) (1990) 267–296
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Computer Science (2013)
14. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. Computer Science **3**(4) (2012) págs. 212–223
15. Zeiler, M.D.: Adadelta: An adaptive learning rate method. Computer Science (2012)