Family History Information Extraction Via Joint Deep Learning

Xue Shi[1], Dehuan Jiang[1], Yuanhang Huang[1], Xiaolong Wang, PhD[1], Qingcai Chen, PhD[1], Jun Yan PhD[1], Buzhou Tang, PhD[1*]

[1]Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology, Shenzhen, GuangDongGuangdong, China

[*]Corresponding author: tangbuzhou@gmail.com

Abstract

Family History(FH) information is very important in the decision-making process of diagnosis and treatment, however the main source of FH data is stored in semi-structured or unstructured format in electronic health records(EHRs), thus need a Natural Language Processing(NLP) system to extract FH form clinical and biomedical text. In the task1 of the BioCreative/OHNLP2018, we adopt Joint Learning to model entity and relation simultaneously. Our best run achieves the F1- score 0.8861 on subtask1 and 0.5527 on subtask2.

Keywords

Entity Recognition; Relation Extraction; Joint Learning; Bi-LSTM-CRF

1. Introduction

FH is very important in the diagnosis and treatment decision-making process, however these information are mainly stored in semi-structured or unstructured format in EHRs. This brings inconvenience and burden to the doctors. So we need a tool to extract FH information. This involves two important technologies: Entity Recognize and Relation Extraction, which are exactly the two subtasks of BioCreative/OHNLP2018 task1.

Entity recognition is one of the most basic and important task in NLP, but due to domain limitation, for example, patient privacy and data confidentiality constraints, the development of it is slowly compared to other domains. In the past few years, machine learning algorithms were used in some shared tasks, such as the Center for Informatics for Integrating Biology & the Beside (i2b2) 2009 [1], 2010 [2], 2012 [3] and 2014 track1 [4] datasets. The machine learning-based methods include support vector machine (SVM), hidden markov model (HMM), structured support vector machine (SSVM) and conditional random field (CRF),etc. And CRF is the most useful among above algorithms. With the improvement of computer computing power and the development of hardware devices, deep learning quickly became popular in many domain and performed very well. Recurrent Neural Network(RNN) is considered to be the natural method to deal with NLP tasks, for it can connect previous information to the present. However, in practice, RNN doesn’t seem to able learn long-term dependencies. Thankfully, Long Short Term Memory networks(LSTM)[5] introduced by Hochreiter & Schmidhuber (1997)solved this problem. And LSTMs are now widely used to get context representation. Now the common method in entity recognize is LSTM-CRF[6],which is proposed by Huang et al.(2015).Later, the researcher thought that the morphological features could be obtained by the representation of characters. Then produced character level LSTM-CRF.

Relation Extraction is a primary task in information extraction.It contain two aspects:Relation type and relation parameters.Relation type indicates what relationship is between entities, and relation parameters refers to the entity that has the relation.The relation with two parameters is called binary relation, which is the common relation. The study of entity relation extraction is mainly concerned with the extraction of binary relations. The research methods are mainly based on classification. The early elation extraction system is based on knowledge base[7], and then statistical-based machine learning methods were gradually favored by researchers[8]. In recent years, deep learning made great progress in many task, for example,limage processing, later the method of deep learning has also been applied to NLP tasks. Liu et al(2013) first use Convolution neural network(CNN) in relation extraction[9]. Then there are many improved versions based on CNN[10,11,12]. Zhang et al,(2015) start try adopt recurrent neural network(RNN) to do relation classification[13]. Zhou et al,(2016) add attention to RNN[14].

The common method to deal with entity reconition and relation extraction via pipeline. Though, pipeline is more flexible to design, pipeline tend to occur error propagation[15], recently, some joint models based on neural networks methods have been proposed.Li et al proposed the first model to incrementally predict entities and relations using a single joint model [15]. Miwa et al adopt LSTM-based model to extraction entity and relation[16].

In this task, we adopt the Joint Learning method to extract FH information.

2. Material and Methods

Our joint learning model main include two part: 1) Bi-LSTM-CRF as the model to recognize entities. and it contain three layers: input layer: take word embeddings and part-of-speech(POS) embedding as input; Bi-LSTM layer: Get the context information representation of sentences; CRF layer: Concat the output of Bi-LSTM as the input of CRF and output a sequence of labels. In Fig.1 the white vertical rectangle denote word embedding and the blue vertical rectangle denote POS embedding. 2) Bi-LSTM to classify the relations, in this part the Bi-LSTM is also sequence structured[17]. The overview architecture of our Joint Learning model is show in Fig.1, 'B-LS' denote 'B-LivingStutas', and 'B-FM' denote 'B-FamilyMember'.

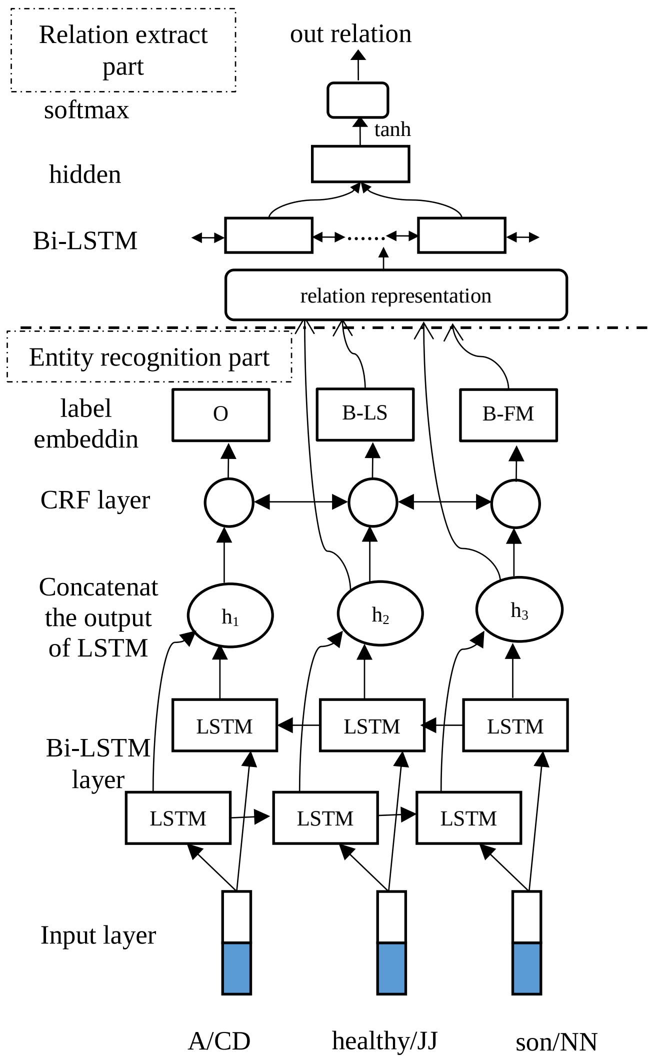

Fig.1 overview architecture of our Joint Learning model

2.1 Dataset

In the OHNLP2018-FH challenge, organizer provide 99 files, we divided them into two parts: (1) 89 files used as a training set; and (2) the remaining 10 files used as a validation set. We split sentence using '\n', because there are only 23 relations cross sentence by using '\n', on the contrary, there will be 339 relations cross sentence by using {'.',':',';'}.After sentence split, there are 449 sentences in training set, and 59 sentences in validation set. For entity recognition, we adopt 'BIO' tagging scheme, 'B' is the beginning of an entity, 'I' mark the token inside an entity, and 'O' mark the other token which are not entities. We joint 'BI' tags and entity type with '-', for example, 'B-FamilyMember' is denote a token is the begin of an entity and the entity's type is 'FamilyMember'.

2.2 Entity recognition

As described above, we modeled entity recognition by character-level Bi-LSTM CRF. Now we introduce each layer one by one in detail.

2.2.1 Input layer

Our input include two part: Word embedding and POS embedding. The token-level representation(word embedding) is usually pre-trained on a large unlabeled corpus, there some available open source can get them, in this task we used the Glove (https://nlp.stanford.edu/projects/glove/).And we use NLTK(https://www.nltk.org/) tool to get the POS of each token, the take the POS to one-hot representation, final get the POS embedding .

2.2.2 Bi-LSTM layer

Bidirectional LSTM (Bi-LSTM)is considered can capture both past and future contexts of words, we use it to get the sentence's every position representation. For given a sentence($x_1x_2...x_i...x_n$) as input, n is the number of words in a sentence. Then produces a sequence of context representation($h_1h_2...h_i...h_n$), where $h_i=[h_{fi},h_{bi}]$ is a concatenation of outputs of forward and backward LSTMs. The forward LSTM take the sentence from start to end as input, and the forward LSTM take the same sentence in reverse as input. Next we will introduce the LSTM unit. It contain there gates: an input gate, a forget gate and an output gate. Input gate controls the proportion of input information transferred to current memory cell; The forget gate controls the proportion of historical information transferred to the current memory cell, and the output gate controls the proportion the information in current memory cell transferred to next state.

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
\tilde{C}_t &= \sigma(W_C \cdot [h_{t-1}, x_t] + b_C) \\
c_t &= f_t * c_{t-1} + i_t * \tilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t * \tanh(C_t),
\end{aligned} \tag{1}
$$

Where σ denote element-wise sigmoid function, i_t is input gate, f_t is forget gate, o_t is output gate, c_t is memory cell, h_t is hidden state, x_t is input. b_g denote bias, W_g denote weight matrices, where $g \in \{i, f, c\}$.

2.2.3 CRF layer

In entity recognition, there are strong dependencies across output labels,for example, B-Observation can not followed by I-LivingStatus. Thus, we use CRF model the output of the whole sentence.

CRF takes sequence $x^{(i)}=(x_1x_2...x_j...x_n)$ as input, and output the label sequence $\hat{y}=(y_1y_2...y_j...y_n)$ which has the max probability. Given a training set $D=\{(x^{(i)},y^{(i)})\,|\,i=1,...,m\}$, all parameters(denote as θ) of CRF are estimated by maximizing the following log-likelihood:

$$L(\theta)=\sum_{i=1}^{m}\log p(y^{(i)}\,|\,x^{(i)},\theta) \qquad (2)$$

Where

$$p(y^{(i)}\,|\,x^{(i)},\theta)=\frac{\exp(\sum_{t=1}^{n}\theta^{T}{}_{y^{(i)}_{t-1}y^{(i)}_{t}}x_t^{(i)})}{\sum_{y\in Y(x^{(i)})}\exp(\sum_{t=1}^{n}\theta^{T}_{y_{t-1}y_t}x_t^{(i)})} \qquad (3)$$

$Y(x^{(i)})$ is the all possible label sequences of $x^{(i)}$

When in test, CRF aim to get the label sequence y^* which has the highest conditional probability:

$$y^*=\underset{y\in Y(x)}{\arg\max}\,p(y\,|\,x,\theta)=\underset{y\in Y(x)}{\arg\max}\,p(y\,|\,h,\theta) \qquad (4)$$

h is the output of Bi-LSTM, and now it is the input of CRF.

2.3 relation extract

In this part we use Bi-LSTM with sequence structure to model relations. We pair-wise the entity which is recognize in entity recognition part, in our task, there are some entity must have not relation, thus we specified the contraints: Let {'FamilyMember'} be a set, and {'Observation','LivingStatus'} be the other set, only pair the entities in different set,entities in same set are not paired. Then takes a pair entity with entity label and the relation type with context between them as input to the Bi-LSTM , and obtains the corresponding representation which is the concatenation of outputs of forward and backward LSTMs. Then the representation via softmax to get the relation type. There is two relation type: 'Yes': If the 'Observation' or 'LivingStatus' is belong to the 'FamilyMember', otherwise, will be the other type 'No'.

2.4 Joint learning of entity recognition and relation extract

We use cross-entropy as the loss function, L_e and L_r denote the loss of entity recognition and relation extract respectively. The loss of our Joint learning is :

$$L=\alpha L_e+(1-\alpha)L_r, 0<\alpha<1 \qquad (5)$$

where α is the bias coefficient. α is larger, the influence of entity recognition is greater, otherwise, the relation extract has greater influence.

2.5 Rule-based post processing

In the entity recognition part, we identify the three kinds of entity: FamilyMember, Observation and Livingstatus, however these information are raw, there need three subtask to do:

1) the FamilyMember's form is diverse, which need to be normalized, we use rules normalize them with dictionary.

2) The FamilyMember has a property : side of family ,which has three value: Paternal / Maternal / NA. We use rules to deal it. Our rules are very sample: First , if the familymember is First-degree relatives, the value of side of family is 'NA' , if it is not First-degree relatives, we find if there is the substring 'maternal family history:' or 'paternal family history:' at the beginning of the sentence, then the values of all familymembers' property can be determined, otherwise, we find if there is indicator near familymember, then the property can be determined, if not ,the property is 'NA'.

3) In OHNLP2018-FH task, Livingstatus has two properties : 'Alive' and 'Healthy', we need quantify them by score. In this step, we also adopt rules to score 'Alive' and 'Healthy'. We first judge if the familymember is alive, then judge if it is healthy.

3. Result

In this task, we did two sets of comparative experiments: 1) Consider the property of FamilyMember as entities, thus the FamilyMember was divided into three parts: Maternal(when the property is maternal), Paternal(when the property is paternal) and FamilyMember(when the property is NA), so there will be five kinds of entities: FamilyMember, Maternal, Paternal, Observation and LivingStatus. 2) Do not split FamilyMember, just recognize three kinds of entity: Familymember, Observation and Livingstatus. The experiment setting is show as in Table1.

Table 1

Hyperparameters chosen for all our experiments

Hyperparameters	value
Dimension of word embedding	50
Token lstm hidden state dimension	100
optimizer	SGD
Learning rate	0.005
Dropout in entity recognize	0.5
Dropout in relation extract	0.3
bias coefficient(α)	0.4/0.5/0.6

The experiment result is show in Table 2(a) and Table 2(b), according to the result, we can conclude: in entity recognition, three tpye entities is higher than five type, however, in relation extract, five type entities is higher than three types. The result in Table 2(a) and Table 2(b) is the best result under its corresponding parameters, we set early stop when the result not improve in 100 epochs. And '-' denote that there is not evaluation in the table.

Table 2(a)

F1-score of entity in two sets of comparative experiment with different bias coefficient

bias coefficient(α)	Validation set		Test set	
	3 type entity	5 type entity	3 type entity	5 type entity
0.4	0.8693	0.8693	-	0.8828
0.5	0.8753	0.8718	0.8852	-
0.6	0.8831	0.8747	**0.8861**	-

Table 2(b)

F1-score of relation in two sets of comparative experiment with different bias coefficient

bias coefficient(α)	Validation set		Test set	
	3 type entity	5 type entity	3 type entity	5 type entity
0.4	0.6020	0.6978	-	**0.5527**
0.5	0.6316	0.6879	0.4534	-
0.6	0.5543	0.6769	0.4356	-

The three runs we submit are: 1) 5 type entity with bias coefficient 0.4; 2) 3 type entity with bias coefficient 0.5; 3) 3 type entity with bias coefficient 0.6.

4. Discussion

In Table 1, we discover that the result of three kinds entities is about 0.01 points higher than five kinds entity. We consider that there are two reasons for this result: 1) the corpus is small, when we split the FamilyMember into three part, there are more smaller sample. 2) For we split sentence by '\n', some sentence are long, and the words imply the property are at the the beginning of the sentence, which is far away from the FamilyMember, so the Paternal or Maternal will be recognize as FamilyMember. In this task, we take the three-category plan, and identify FamilyMember's property by rules. However, we think this problem can also be solved by adding attention mechanism into LSTM network, though we did not do this experiment.

5. Conclusion

In this task, we first adopt Joint Learning to get entity and relation pairs, then seeking the characteristics of the corpus, we use rules to do post processing. The result show that the entity recognition at coarse-grained divided entity(3 type entity) is better than fine-grained divided entity(5 type entity) in small data set, somehow the relation extract is better in fine-grained divided data, the reason maybe is that fine-grained divided entity provide property feature to relation extract.

6. References

1. Uzuner Ö, Solti I, Cadag E. Extracting medication information from clinical text. J Am Med Inform Assoc. 2010;17:514‐8.

2. Tang B, Cao H, Wu Y, Jiang M, Xu H. Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. BMC Med Inform Decis Mak. 2013;13:1.

3. Xu Y, Wang Y, Liu T, Tsujii J, Eric I, Chang C. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. J Am Med Inform Assoc. 2013;20:849–58.

4. Liu Z, Chen Y, Tang B, Wang X, Chen Q, Li H, Wang J, Deng Q, Zhu S. Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. J Biomed Inform. 2015;58:S47–52.

5. Hochreiter S, Schmidhuber J. LSTM can solve hard long time lag problems[C]//Advances in neural information processing systems. 1997: 473-479.

6. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.

7. Aone C, Ramos-Santacruz M. REES: a large-scale relation and event extraction system. In: Proceedings of the 6th Conference on Applied Natural Language Processing.Stroudsburg, PA, USA: Association for Computational Linguistics, 2000. 76−83.

8. Nadeau D, Sekine S. A survey of named entity recognition and classification. Lingvisticae Investigationes, 2007, 30(1):3−26

9. Liu C Y, Sun W B, Chao W H, et al. Convolution neural network for relation extraction[C]//International Conference on Advanced Data Mining and Applications. Springer, Berlin, Heidelberg, 2013: 231-242.

10. Zeng D, Liu K, Lai S, et al. Relation classification via convolutional deep neural network[C]//Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 2014: 2335-2344.

11. Nguyen T H, Grishman R. Relation extraction: Perspective from convolutional neural networks[C]//Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing. 2015: 39-48.

12. Santos C N, Xiang B, Zhou B. Classifying relations by ranking with convolutional neural networks[J]. arXiv preprint arXiv:1504.06580, 2015.

13. Zhang D, Wang D. Relation classification via recurrent neural network[J]. arXiv preprint arXiv:1508.01006, 2015.

14. Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016, 2: 207-212.

15. Q. Li, H. Ji, Incremental joint extraction of entity mentions and relations., in:Proceedings of the 52rd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 402 – 412.

16. M. Miwa, M. Bansal, End-to-end relation extraction using lstms on sequences and tree structures, in: Proceedings of the 54rd Annual Meeting of the Association for Computational Linguistics, 2016.

17. Zheng S, Xu J, Bao H, et al. Joint learning of entity semantics and relation pattern for relation extraction[C]//Joint european conference on machine learning and knowledge discovery in databases. Springer, Cham, 2016: 443-458.