A CRD-WEL System for Chemical-disease Relations Extraction

Zhenchao Jiang[1], Liuke Jin[1], Lishuang Li *, Meiyue Qin, Chen Qu, Jieqiong Zheng, Degen Huang

School of Computer Science and Technology, Dalian University of Technology
Dalian116024, Liaoning, China

jzc_nlp@163.com;
lkjin_email@163.com;
*lilishuang314@163.com;

Abstract. As one task of the BioCreative V competition, the chemical-disease relations (CDR) include two subtasks: DNER and CID. We participated in this track and designed two separate systems for each subtask. The CRD-WEL system consists of two subsystems: CRD-DNER and WEL-CID. For DNER, the CRD-DNER system is proposed, which is a combined system for disease named entity recognition based on shallow and deep models. For CID, WEL-CID system uses a novel word embedding model and logistic regression classifier to extract Chemical-induced Diseases from text.

Keywords. Disease named entity recognition; Shallow and deep models; Chemical-induced Disease; Word Embedding; Logistic Regression

1 Introduction

Chemicals, diseases, and their relations play central roles in many areas of biomedical research and healthcare such as drug discovery and safety surveillance, therefore, they are the most searched topics by PubMed users worldwide [1–3]. Automatic extraction of chemical-disease relations (CDR) from unstructured free text into structured knowledge has become an important theme for bioinformatics databases such as the Comparative Toxicogenomics Database (CTD) [4].

For DNER task, considering both shallow model, i.e. conditional random field (CRF), and deep model, i.e. recurrent neural network (RNN), have been previously used in the NER task, the two models are combined based on the probabilities, and we also adopt dictionary pat-

[1] These authors contributed equally to this work; * the corresponding author

tern matching to improve the recall of disease mentions. Finally, we directly use the DNorm tools provided by NCBI [5] to return normalized disease concept identifiers.

As for CID task, feature based methods and kernel based methods are widely used for relation extraction. For feature based methods, most previous works used one-hot coding, which fails to capture the semantic meaning of words, and for kernel based methods, most kernels are highly depended on the parsing. We previously presented an instance representation architecture for Protein-Protein Interaction extraction, which takes advantage of word embeddings trained using Skip-gram and skeleton features [6]. Therefore, in this paper, we propose a novel word embedding training model and integrate word embeddings into Chemical-induced Disease (CID) task.

2 CRD-DNER System

Fig. 1 shows the workflow of CRD-DNER system, which consists of **C**RF model, **R**NN model and **D**ictionary pattern matching.

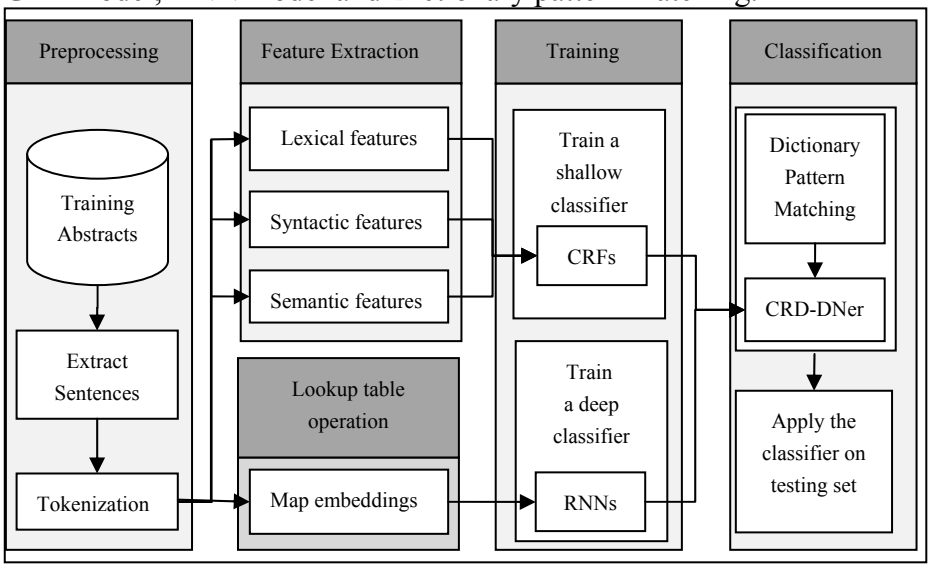

Fig. 1. CRD-DNER system workflow

2.1 Data Representation

Shallow Model.

The features used by the shallow model can be categorized into lexical, syntactic and semantic as shown in Table 1.

Lexical and syntactic features can be obtained from the word appearance and NLP tools e.g. GENIA tagger. We use Brown Cluster algorithm[7] to acquire 512 clusters, and each cluster is assigned a binary representation based on Huffman coding [8].

Table 1. Features used in the CRF model

category	Feature name	Description
lexical	Word	Original word and its context
	Shape	Capital letters, lowercase letters and digits are replaced with 'A', 'a' and '0', respectively; other letters are replaced with '_'.
	morphology	1. Digits in the current word are replaced by a '*'and capital letters are lower cased. If no digits, it is replaced by "no". 2. Each letter in the word is replaced by '-', except 5 vowels.
	prefix	2-4 character prefixes
	suffix	2-4 character suffixes
	orthographic	A total of 29 orthographic features are used in the model
	length	The length of the current word which may be 1, 2, 3-5 or >=6.
syntactic	Pos	The three features are obtained by GENIA tagger.
	chunk	
	stem	
semantic	Brown cluster	6-9 binary prefixes of the Huffman coding.

Deep Model.

In our recurrent neural network architecture, we use the distributed representations of words trained by an improved Word2Vec tool [9]. A real-valued embedding vector is associated with a word, and all the real-valued vectors are trained with 1000 abstracts from the training set and development set. In our experiments, the vector dimension is set to 200. After that, we can map each word in our corpora to an embedding and initialize the word lookup table with the embeddings. And in order to capture short-term temporal dependencies in a sentence, we use a word-context window, and then concatenate all the vectors in the window to make up the raw input of the deep model.

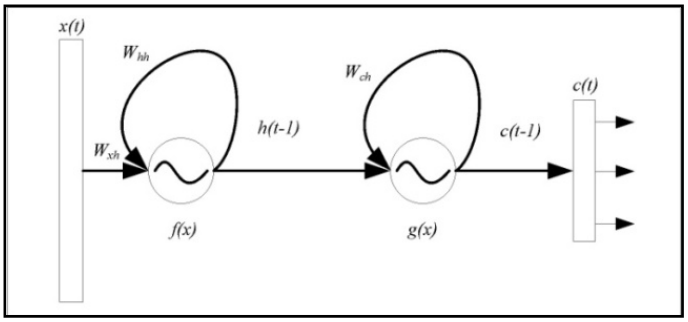

Fig.2. Recurrent neural network architecture

2.2 Training

In the deep RNN model, we present an improved RNN architecture which can not only maintain a copy of hidden layer but also record the probability of output layer as shown in Fig.2. When computing hidden neurons, we take the same action with Elman-type RNN[10] and use sigmoid function as the activation function as shown in equation (1). Different from Jordan-type RNN [11], the results of output layer from last node are inputted into the current output layer with equation (2).

$$h(t) = f(x(t) \cdot W_{xh} + h(t-1) \cdot W_{hh} + b_h) \quad (1)$$

$$c(t) = g(h(t) \cdot W_{hc} + c(t-1) \cdot W_{cc} + b_s) \quad (2)$$

2.3 Combination

We compute the information entropy respectively ($H(p)_{shallow}$ and $H(p)_{deep}$) with equation (3), and select the label predicted by the model which has a lower value as the correct label.

$$H(p) = -\sum p(y) log p(y) \quad (3)$$

Meanwhile, a dictionary is built from Comparative Toxicogenomics Database (CTD) [12]. At each time, after using this combined model to extract disease names from an abstract, we also extract new disease names from the abstract with dictionary pattern matching.

2.4 Normalization with DNorm

DNorm is a technique to find the best name from a controlled vocabulary such as MeSH for a given mention. It uses a regression model

learned directly from the training data to score each name in the controlled vocabulary against the mention provided as query and returns the top ranked name [13]. We employ the same ranking method with [5] to normalize each mention to find the disease concepts in the lexicon.

3 WEL-CID system

The word embedding and logistic regression based CID (WEL-CID) system contains the following five steps:

3.1 Preprocess

Due to the great variation of biological names in biomedical text, appropriate tokenization is an important preprocessing step for learning biomedical word embeddings. Experimental results show that tokenization can significantly affect the retrieval accuracy and appropriate tokenization can significantly improve the performance [14], which inspires us to take tokenization seriously in our method. The aim of tokenization is to tokenize the sentence into atomic units. E.g., "(IL-2)" is composed of "(", "IL-2" and ")". To achieve this, first tokenize a sentence using space and characters in "" _ * ; / ! ? = } { ~ ` # $ ^ & ” “ \ | ± °", then for each token *t*, if *t* contains only one character of ", . ' : ," which appears at the end of *t*, then strip this character. Finally, strip the brackets if t is bracketed.

After tokenization, we use DNorm and tmChem to recognize and normalize the disease mentions and chemical mentions respectively, and use GDEP to analyze the text to obtain syntactic chunks and dependencies.

3.2 Word Embedding Training

Skip-gram, CBOW [15, 16] and many other neural embeddings use a sliding window of size k around the target word w, $2k$ contexts are produced: the k words before and the k words after w. A context window of tokens may miss some important factors. After we preprocess the text, we obtain the stems, chunks and entities, which are utilized to train word embeddings (taking the sentence " Phenobarbital

induced dyskinesia in a neurologically impaired child. " for example as shown in Fig. 3) .

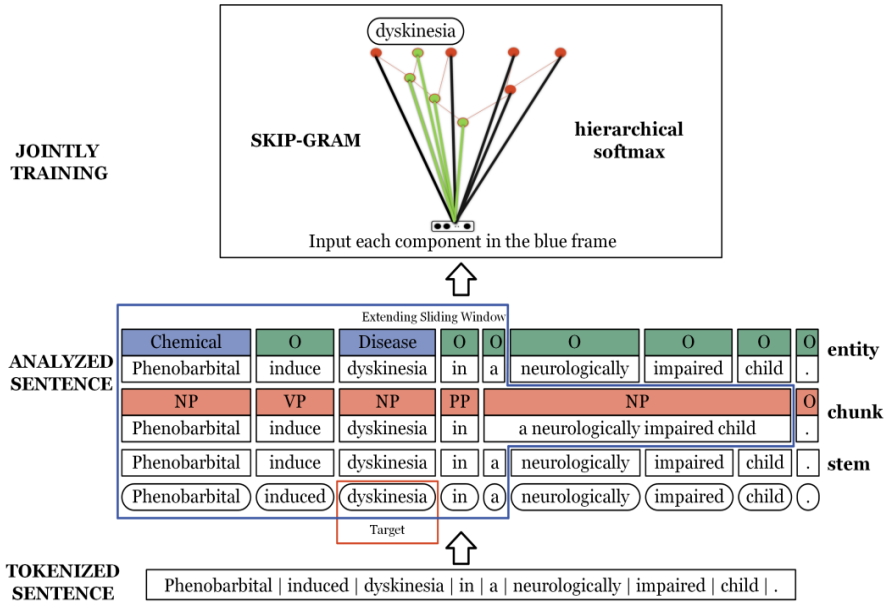

Fig. 3. A novel word embedding training model for biomedical text mining, taking advantage of stem, chunk and biomedical entity.

3.3 Feature Extraction

Considering CID relation across sentences, five kinds of features are used in our system:

Chemical mention: the name of the chemical mention.

Disease mention: the name of the disease mention.

Surrounding words: the left m words before and after the two mentions.

Sentence words: the words in the two sentences where the two mentions locates in.

Dependency path: the words in the dependency path between the two mentions. Empty if across sentences.

The extracted features are converted into numerical matrices by looking up table obtained in section 3.2. Four kinds of vector

composition strategies are used: max, min, sum and mean, and then concatenate the vectors to obtain the final input vectors.

3.4 Principle Component Analysis

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. Our method uses PCA to downsize the input vectors and obtain better representation. Finally, our method uses logistic regression to classify each CID mention pair.

3.5 Post Process

So far we have classified each mention pair into positive or negative, however there exists among those pairs. For example, for a pair of Chemical and Disease, there are more than one corresponding mention pairs, some of which are classified as positive while some are negative.

First, if the distance between the two mentions is short, and no other mentions between them, and there exists one of the following words between them, "effect, induce, cause, produce, enhance, due to, administrate, from, sustain", then the CID relation is considered to be positive.

Second, we discard distant mention pairs. For each CID relation, we only consider the nearest n corresponding mention pairs, which effectively increases the precision of the CID result.

Third, in the retained n pairs, if more than one mention pair is classified as positive, then the CID relation is considered to be positive.

4 Experiments and Discussion

4.1 DNER task

In our submissions, we submit 3 runing models as shown in Table 2, i.e. the sole CRF model, CRF with dictionary and our CRD-DNER system. When we evaluated the three models on the development set, the CRD-DNER system can achieve the best performance of 80.43% F-score. When we evaluate them on the test set, the CRF with dictionary outperfoms the other two models by 0.75% and 0.71%.

Table 2. Performance meaurements for each model

Test set	Models	Recall (%)	Precision (%)	F-score (%)
development	CRF	76.66	83.75	79.97
	CRF+Dictionay	77.95	81.63	79.75
	CRF+RNN+Dictionary	78.92	82.00	**80.43**
test	CRF	83.93	72.48	77.79
	CRF+Dictionay	81.76	75.55	**78.54**
	CRF+RNN+Dictionary	79.74	76.01	77.83

4.2 CID task

We conduct a group of experiments on the development set. First, we only use the first and second post process procedure. From Table 3 we find that as the value of *n* increases, the precision decreases and the

Table 3. A comparison of *n*, without applying the third post process strategy.

n	Precision (%)	Recall (%)	F-score (%)
3	61.8	20.5	30.7
5	53.1	29.4	37.9
7	53.7	38.0	44.5
9	49.9	40.7	44.5
11	45.6	35.1	39.7

Table 4. A comparison of *n*, with applying the third post process strategy.

n	Precision (%)	Recall (%)	F-score (%)
7	54.1	51.7	52.9
9	51.5	54.4	52.9
11	47.4	54.1	50.5

recall increases, until *n*=9. The best group, *n*=9, F-score 44.5%, is much higher than when *n*=3. This is to be expected, because the nearer the mentions are, the more likely that the mention pair can be classified correctly. Therefore, the precision, recall and F-score are all sensitive to n, and we think n is an important parameter to optimize the system.

Next we use the third post process procedure. From Table 4, we can find that the third post process strategy can highly improve the performance, the best group, *n*=9, F-score 52.9%, is 8.4% higher than that in Table 3. Therefore, the third post process strategy is highly recommended for CID task.

Note that we have also conduct other groups of experiments, e.g., using Support Vector Machine (SVM) instead of logistic regression, using different word embedding models, normalization strategies, regularization strengths, and feature sets. However, these variations do not influence the performance as much as the post processing procedures do, and some of these variations even decrease the F-score, e.g., logistic regression performs better than SVM in our experiments. Counting all these factors, our best F-score on development set is 54.2%.

5 Acknowledgment

The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China under Nos. 61173101, 61173100.

REFERENCES

1. Doğan, R. I., Murray, G. C., Névéol, A., and Lu, Z. "Understanding PubMed user search behavior through log analysis." Database (2009): bap018.
2. Lu, Z. "PubMed and beyond: a survey of web tools for searching biomedical literature." Database (2011): baq036.
3. Névéol, A., Dougan, R. I., and Lu, Z. "Semi-automatic semantic annotation of PubMed queries: a study on quality, efficiency, satisfaction." Journal of biomedical informatics(2011): 310–318.
4. Davis, A. P., Grondin, C. J. et al. "The Comparative Toxicogenomics Database's 10th year anniversary: update 2015." Nucleic acids research (2015): 914–920.
5. Leaman, R., Doğan, R. I. and Lu Z. "DNorm: Disease name normalization with pairwise learning to rank." Bioinformatics (2013): 2909–2917.
6. Li, L., Jiang, Z., and Huang, D. (2014). "A general instance representation architecture for protein-protein interaction extraction."Paper presented at the 2014 {IEEE} International

Conference on Bioinformatics and Biomedicine, Belfast, United Kingdom, November 2-5, 2014.
7. Peter, F. B., Peter, V. D., Robert, L. M., Vincent, J. D., and Jenifer, C. L. “Class-Based n-gram Models of Natural Language.”Computational linguistics 18.4 (1992): 467-479.
8. Lu, Y., Yao, X., Wei, X., et al. “WHU-BioNLP CHEMDNER System with Mixed Conditional Random Fields and Word Clustering.”Paper presented at the Fourth BioCreative Challenge Evaluation Workshop. 2013, 2: 129-134.
9. Mikolov, T., Chen, K., Corrado, G., and Dean, J. “Efficient Estimation of Word Representations in Vector Space.”arXiv preprint arXiv:1301.3781 (2013).
10. Elman, J. L. “Finding Structure in Time,”Cognitive science 14.2 (1990): 179-211.
11. Jordan, M. I. "Serial order: A parallel distributed processing approach." Advances in psychology 121 (1997): 471-495. .
12. Davis, A. P., Grondin, C. J., Lennon-Hopkins, K. et al. “The Comparative Toxicogenomics Database’s 10th year anniversary: update 2015,”Nucleic acids research 43.D1 (2015): D914-D920.
13. Leaman, R., Khare, R., and Lu, Z. “NCBI at 2013 ShARe / CLEF eHealth Shared Task : Disorder Normalization in Clinical Notes with DNorm,”Radiology 42.21.1 (2011): 1-941.
14. Jiang, J., and Zhai, C. "An empirical study of tokenization strategies for biomedical information retrieval." Information Retrieval (2007): 341–363.
15. Mikolov, T., Chen, K., Corrado, G., and Dean, J. "Efficient Estimation of Word Representations in Vector Space." arXiv preprint arXiv(2013).
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. "Distributed representations of words and phrases and their compositionality." Advances in Neural Information Processing Systems (2013):3111–3119.