Integrating Word Sequences and Dependency Structures for Chemical-disease Relation Extraction

Huiwei Zhou[1], Yunlong Yang[1], Zhuang Liu[1], Zhe Liu[2], Yahui Men[2]

[1] School of Computer Science and Technology, [2] School of Life Science and Medicine
Dalian University of Technology, Dalian 116024, Liaoning, China
zhouhuiwei@dlut.edu.cn, {SDyyl_1949,zhuangliu1992,
menyahui}@mail.dlut.edu.cn, dlutliuzhe@163.com

Abstract. Understanding chemical-disease relations (CDR) from biomedical literature is important for biomedical research and chemical discovery. This paper uses a *k*-max pooling convolutional neural network (CNN) to exploit word sequences and dependency structures for CDR extraction. Furthermore, an effective weighted context method is proposed to capture semantic information of word sequences. Our system extracts both intra- and inter-sentence level chemical-disease relations, which are merged as the final CDR. Experiments on the BioCreative V CDR dataset show that both word sequences and dependency structures are effective for CDR extraction, and their integration could further improve the extraction performance.

Keywords: CDR extraction, CNN, word sequences, dependency structures.

1 Introduction

Extracting chemicals, diseases and their relationships are significantly important to biomedical research and healthcare [1]. Manual annotating chemical-disease relations (CDR) from the vast amount of published biomedical literature is expensive and time-consuming, and is impossible to keep up-to-date. Many automated CDR extraction methods have been proposed. To promote CDR research, the BioCreative V CDR task [2] provides a public dataset as a platform for comparing different methods. The task includes two subtasks：(i) disease named entity recognition and normalization (DNER) and (ii) chemical-induced diseases (CID) relation extraction. In this paper, we focus on the CID subtask with both intra- and inter-sentence levels.

Existing studies on CDR extraction contain rule-based [3] and machine learning-based [4-7] methods. Rule-based methods could make full use of syntactic information and have achieved good performance. But the rules are hard to develop to a new dataset. As for machine learning-based relation extraction, feature-based and kernel-based methods are widely used. Feature-based methods [4-6] focus on extracting effective features. However, the features are one-hot representations, which could not capture deep semantic information. Kernel-based methods [7] define a tree kernel over shallow parse tree representations of sentences, which is also hard to capture deep syntactic structure information.

With the development of neutral networks, some studies begin to exploit deep semantic information for relation extraction. Zhou et al. [7] simply adopt a long short-term memory (LSTM) model [8] and a convolutional neural network (CNN) model [9] to get semantic representations of surface sequence, and have achieved success for CDR extraction. CNN shows its superiority to other neutral networks on relation extraction task [9].

This paper develops a multiple layer CNN model to integrate sequences and dependency structures for CDR extraction. To capture more effective semantic and syntactic information, we use the *k*-max pooling [10] instead of the traditional max pooling. Besides, we propose an extended context representation inspired by Vu et al. [11]. The major difference between our method and Vu et al. [11] is that we concatenate the different context representations by different weights rather than concatenating them equally. Both intra- and inter-sentence level CDR are investigated in this paper to improve the extraction performance. Experiments on the BioCreative V CDR dataset demonstrate the effectiveness of our method. In the following section, we review the literature related to this paper from two aspects: CDR extraction and CNN for relation extraction.

2 Related Work

2.1 CDR Extraction

Lowe et al. [3] develop rules by manually identifying the key words that indicate the CDR. Their system is evaluated on the BioCreative V CDR Task, and achieves 60.75% F-score using gold-standard entities, 52.20% F-score using entities identified by their entity recognizer.

Feature-based methods focus on designing rich features. Gu et al. [5] use effective linguistic features to extract CDR with maximum entropy models. They achieve 58.3% F-score on the BioCreative V test data using gold-standard entities. Xu et al. [4] employ different drug-side-effect resources to generate knowledge-based features for both sentence-level and document-level CDR extraction, and achieve 67.16% F-score using gold-standard entities. Pons et al. [6] use rich prior knowledge features and achieve 70.2% F-score using gold-standard entities.

Kernel-based methods are effective for capturing syntactic structure information. Zhou et al. [7] exploit a shortest dependency path (SDP) tree kernel to capture the predicate-argument relations, which could achieve 50.11% F-score using gold-standard entities. Zhou et al. [7] also use two categories of neural networks, LSTM and CNN, to capture deep semantic representations. To further integrate lexical and syntactic information, two neural models are combined with feature-based model and kernel-based model by a linear combination respectively. Among different neural network models employed for relation extraction [9,12], CNN is superior for relation extraction to the others.

2.2 CNN for Relation Extraction

Nguyen and Grishman [9] first employ CNN for relation extraction. To avoid the noise that originates from the feature extraction process, Zeng et al. [12] propose a piecewise CNN, which divides the convolution results into three segments based on the positions of the two entities, and returns the maximum value in each segment instead of a single maximum value over the entire sentence. Inspired by their work, we apply CNN model to capture both the word sequence-based representations and the dependency-based representations for CDR extraction.

3 Methods

The architecture of our system is shown in Fig. 1, which consists of a training phase and a testing phase. In the training phase, we construct the intra- and inter-sentence level instances from the training data. For intra-sentence level instances, we learn a sequence-based model and a dependency-based model by CNN respectively. For inter-sentence level instances, we learn a sequence-based model by CNN. In the testing phrase, the three models are applied to extract CDR respectively. And the predicted results of the three models are merged as the final results.

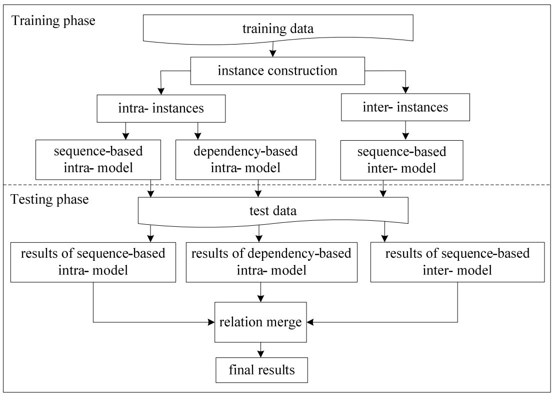
Training phase
training data
instance construction
intra- instances
inter- instances
sequence-based intra- model
dependency-based intra- model
sequence-based inter- model
Testing phase
test data
results of sequence-based intra- model
results of dependency-based intra- model
results of sequence-based inter- model
relation merge
final results

Fig. 1. Architecture of our system.

3.1 Convolutional Neural Network

The architecture of our CNN model are shown in Fig. 2. It consists with four main layers:

(i) the vector representation layer. The vector representation layer embeds each word in a sentence into a low dimensional space. Consider an input of word vector sequence $x=\{x_1,x_2,...,x_n\}$, where $x_n \in \mathbb{R}^d$ is a *d*-dimensional word vector. Then the word vector sequence is fed into the convolution layer.

(ii) the convolution layer. The convolution operation with a filter $w \in \mathbb{R}^{h\times d}$ can be expressed as $\mathrm{c}_i = f\,(w \bullet x_{i:i+h-1}+b)$, where *h* is a window size, *b* is a bias term, *f* is a non-linear function such as the hyperbolic tangent and $x_{i:j}$ refer to the subsequence of words from x_i to x_j . The filter is used to each possible window of words in the sequence $x=\{x_1, x_2, ..., x_n\}$ to produce a feature map: $\mathbf{c}=[c_1, c_2, ..., c_{n-h+1}]$ with $\mathbf{c} \in \mathbb{R}^{n-h+1}$. In our model, multiple filters with different window size are apply to obtain multiple features.

(iii) the *k*-max pooling layer. We use a *k*-max pooling operation instead of the max-pooling to take the *k* highest values as the features. The order of the *k* values in the sequence corresponds to their original order in $\mathbf{c}$. In particularly, when $k=1$, the pooling operation becomes the max pooling operation.

(iv) the softmax output layer. The *k*-max feature vector $z=[c_{11}, c_{12}, ..., c_{1k}, ..., c_{mk}] \in \mathbb{R}^{m*k}$ is fed to a softmax layer to perform classification in the end. Note that here we have *m* filters and each filter select the *k* highest values during the pooling.

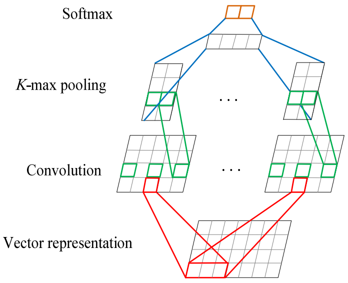

Fig. 2. The architecture of CNN model.

3.2 Intra-sentence Level CDR Extraction

To explore deep semantic and syntactic information behind CDR pairs of intra-sentence level instances, we learn word sequence-based representations and dependency-based representations by CNN.

Word Sequence-based Representations. The following representation methods based on word sequences are introduced and compared in this paper.

Word This method inputs the word sequences between chemical and disease entities into CNN to capture semantic representations of CDR pairs. The dimension of word representation $x_w \in \mathbb{R}^{d_1}$ is d_1. We regard this method as our baseline.

Word-position Besides the word sequences, this method also inputs position tags of the word sequences. The relative distances from the current word to the two entities are transformed into representations. Then the representations of each word and its position tags are concatenated to form a vector representation $x_w, x_p \in \mathbb{R}^{d_1+d_2*2}$.

Word-context The **Word** method only includes internal context between chemical and disease entities. Motivated by Vu et al. [11], we introduce external context for relation extraction. The sentence is split into three disjoint regions based on the two entities: the left external context, the middle internal context and the right external context. Not only focusing on the middle internal context but also not ignoring the other external contexts, we use two contexts (1) *L*: a combination of the left external context, the left entity and the middle internal context; (2) *R*: a combination of the middle internal context, the right entity and the right external context. The difference between our model and Vu et al. [11] is that we do not connect the two contexts equally. Instead, we use the weight α， $\beta \in [0,1]$($\alpha+\beta=1$) to control the connection of the two contexts as follows:

$$z = [\alpha \bullet L_{11}, \alpha \bullet L_{12}, ..., \alpha \bullet L_{mk}, \beta \bullet R_{11}, \beta \bullet R_{12}, ..., \beta \bullet R_{mk}] \quad (1)$$

The weight α， β are the parameters that the network need to learn, which are both initialized as 0.5 at first. In order to compare the traditional connection method with the weighted concatenation, we name them as **Word-context** and **Word-weighted-context** respectively.

Dependency-based Representations. SDP is the shortest path linking the two entities in dependency tree. Taking Sentence 1 as an example, a chemical entity is denoted by wave line and three disease entities are denoted by underline. The chemical entity "*Lithium*" is associated with the three disease entities.

Sentence 1: *Lithium* also caused *proteinuria* and systolic *hypertension* in absence of *glomerulosclerosis*.

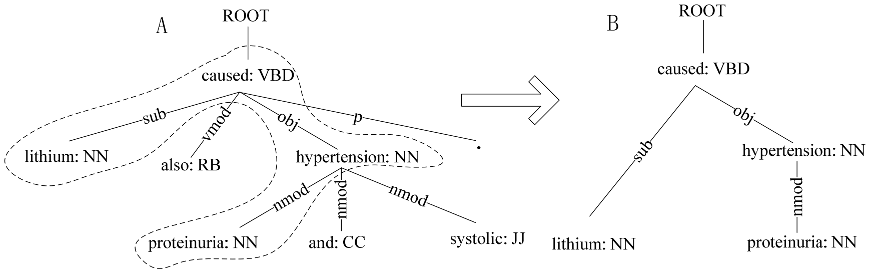

Fig. 3. Shortest dependency path tree (SDPT). (A) the fragment of dependency tree for Sentence 1. (B) shortest dependency path.

For the fragment of dependency tree (all words in Sentence 1 are transformed to lowercase) shown in Fig. 3A, SDP of the candidate "*lithium*" and "*proteinuria*" is shown in Fig. 3B. The following input methods are adopted to learn dependency-based representations.

SDP-word This method inputs a sequence of words in the SDP into CNN to capture dependency semantic representations behind SDP. Note that the sequence follows the left-to-right order in SDP as shown in Fig. 4A.

SDP-dep Compared with **SDP-word**, this method inputs the sequence consists both words and dependency relations of SDP as shown in Fig. 4B. The dimensions of word representations $x_w \in \mathbb{R}^{d_1}$ and relation representations $x_r \in \mathbb{R}^{d_1}$ are both d_1.

SDPSeq-word Compared with **SDP-word**, this method also inputs a word sequence of SDP. However, the sequence follows the natural order of words in the sentence as shown in Fig. 4C. We consider that this order could reflect the actual semantic information in context.

SDPSeq-dep This method also inserts the dependency relations into the **SDPSeq-word** as shown in Fig. 4D.

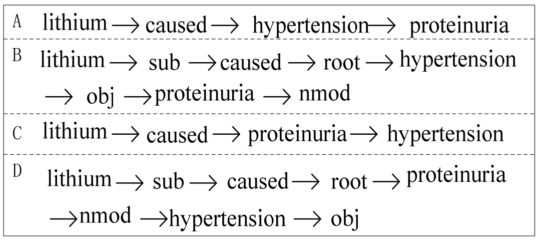
A lithium → caused → hypertension → proteinuria
B lithium → sub → caused → root → hypertension → obj → proteinuria → nmod
C lithium → caused → proteinuria → hypertension
D lithium → sub → caused → root → proteinuria → nmod → hypertension → obj

Fig. 4. SDP sequences. (A) **SDP-word**. (B) **SDP-dep**. (C) **SDPSeq-word**. (D) **SDPSeq-dep**.

3.3 Inter-sentence Level CDR Extraction

To reduce the inter-sentence level instances, the following heuristic filtering rules are applied on both training and testing datasets:

- Only the entities that not involved in any intra-sentence level are considered at the inter-sentence level.
- The sentence distance between two mentions in an instance should be less than 3.
- If there are multiple mentions that refer to the same entity, choose the pair of the inter-sentence level chemical and disease mentions in the nearest distance.

After that, the sequence-based intra- model learning method is applied to learn the inter- model based on these inter-CDR instances. **Word-context** representation methods are not employed since the two entities are not in one sentence.

3.4 Intra- and Inter- Level CDR Merge

The intra- and inter-sentence level extraction results are merged as the final relations. The CDR extracted by each level model are regarded as the true positives. Thus, the final CDR are consists with the following two parts:

- All the CDR extracted by the intra-sentence level model.
- The CDR extracted by the inter-sentence level model only if there are no intra-sentence level CDR find in the abstract.

3.5 Post-processing

To further pick more likely CDR and improve the performance, some rules are applied to help extract relations.

- **Focused rules for the post-processing**. If there are no CDR found in the abstract, all the chemicals in the title are associated with all the diseases in the entire abstract. When no chemical in the title, the chemical in the most occurrences number in the abstract is chosen to associate with all the diseases in the entire abstract.

- **Hypernym filtering for the post-processing**. There are hypernym or hyponym relationship between concepts of diseases or chemicals. However, the goal of the CID subtask aims to extract the relationships between the most specific diseases and chemicals. Therefore, we determine the hypernym relations based on the Mesh-controlled vocabulary [13] following the post-processing in Gu et al. [5]. Then we remove the positive instances that involve entities which are more general than other entities already extracted as the positive ones.

4 Experiments and Discussion

Dataset. Experiments are conducted on the BioCreative V CDR Task corpus. This corpus contains 1500 PubMed articles: 500 each for the training, development and test set. We combine the training and the development sets into the final training set and randomly select 20% of this set as the development set. We test our system on the test set with the golden standard entities. All sentences in the corpus are preprocessed with GENIA Tagger[1] and Gdep Parser[2] to get lexical information and dependency trees, respectively. The evaluation of CDR extraction is reported by official evaluation toolkit[3], which adopts Precision (*P*), Recall (*R*) and F-score (*F*) to measure the performance.

Hyperparameter Settings. For all the experiments below, 100 filters with the window size 3, 4 and 5 respectively are used in our system. In our experiments, we use Word2Vec tool[4][14] to pre-train word representations on the datasets (about 8868MB) downloaded from PubMed[5].The dimension of word embeddings, dependency type embeddings and position embeddings are 100, 100 and 30 respectively.

4.1 Effects of the *k*-max pooling

In this section, we compare the performance of the *k*-max pooling with the max pooling for CDR extraction. Several input methods are selected to learn representations. Table 1 shows the performance of different input methods with different *k*. We vary *k* from 1 to 4.

From Table 1, we can see that the trends of the three methods are similar. When we increase *k* from 1 (the max pooling) to 2, the performance of all methods is improved. This indicates that the *k*-max pooling could capture more effective information and produce deep semantic representations than the max pooling method. However, when *k* increases to 2 and 4, the performance drops. The reason may be that too much noise features are select during the pooling, which could harm model performance. We set $k = 2$ in the following experiments.

[1] http://www.nactem.ac.uk/GENIA/tagger/

[2] http://people.ict.usc.edu/~sagae/parser/gdep

[3] http://www.biocreative.org/tasks/biocreative-v/track-3-cdr/

[4] https://code.google.com/p/word2vec/

[5] http://www.ncbi.nlm.nih.gov/pubmed/

Table 1. Performance of different *k* values.

Methods	*k*=1 (%)			*k*=2 (%)			*k*=3 (%)			*k*=4 (%)		
	P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Word	47.41	57.60	52.01	49.78	54.50	**52.04**	48.99	54.50	51.60	51.70	51.31	51.51
Word-position	52.68	49.81	51.89	55.19	49.81	**52.36**	50.65	50.84	51.30	55.74	46.90	51.61
SDP-word	55.36	51.88	53.56	51.78	55.91	**53.77**	53.26	52.81	53.04	53.20	52.91	53.06

4.2 Performance of the Intra-sentence CDR extraction

In this section, we evaluate the word sequence-based and dependency-based representations for the intra-sentence CDR extraction.

Performance of the word sequence-based representations. The detailed performances of the word sequence-based methods are summarized in Table 2. From the results, we can see that:

- The **Word** method with only the word sequence has achieved an acceptable result, which demonstrates the superiority of CNN for relation extraction.
- When the position embeddings are added to the word sequence, the performance of **Word-position** is improved. This indicates that encoding the relative distances to the entity pairs is effective for CDR extraction.
- The **Word-context** method shows a better result than the **Word** method. The reason may be that the trigger words which indicate the CID relation would occur not only in the middle contexts but also in the left or the right contexts.
- The **Word-weighted-context** improves the performance further. It is believed that given different weights to the contexts could reduce the noise data, and result in higher F-score. The best performance is obtained with α=0.589 during the training process.

Table 2. Performance of word sequence-based representations.

Methods	***P*** **(%)**	***R*** **(%)**	***F*** **(%)**
Word	49.78	54.50	52.04
Word-position	55.19	49.81	52.36
Word-context	57.40	50.18	53.55
Word-weighted-context	53.98	53.47	**53.72**

Performance of the dependency-based representations. Table 3 shows the performance of the dependency-based methods on the CDR extraction. From this table, we can see that:

- The **SDP-word** get a better result than the **Word** in Table 2. Thus, it can be seen that SDP could capture the most direct semantic representation connecting the two entities and provide the strong hints for the relation extraction.
- When we add the dependency type in the **SDP-word**, the F-score of the **SDP-dep** improves to 53.90%. The dependency type can reflect the syntactic relation between two words, which lead to improvement in extraction precision.
- However, the **SDPSeq-dep** fails to catch up the **SDP-dep**, and the **SDPSeq-word** fails to catch up the **SDP-word** similarly, which suggest that the natural order of

words may lose the structure information and is hard for CNN to capture the semantic representations.

After getting both the sequence-based representations and dependency-based representations, we combine the best sequence-based (**Word-weighted-context**) and dependency-based (**SDP-dep**) models as the final intra-sentence level model. Then for each intra-sentence instance x in the test set, the predicted relation label y is calculated by $y = \arg\max_{l \in \{0,1\}} (P_{seq}(l \mid x) + P_{dep}(l \mid x))$, where $P_{seq}(l \mid x)$ and $P_{dep}(l \mid x)$ represent the predicted probabilities of the sequenced-based and dependency-based models with the relation label $l \in \{0,1\}$. This method is called **Combination.** The result reaches 55.15% F-score as shown in Table 3. This indicates that both the sequence-based model and the dependency-based model have their own advantages and could capture different information for CDR extraction. Their combination could further improve the performance.

Table 3. Performance of dependency-based representations.

Methods	***P*** **(%)**	***R*** **(%)**	***F*** **(%)**
SDP-word	51.78	55.91	53.77
SDP-dep	54.74	53.10	53.90
SDPSeq-word	53.27	51.88	52.57
SDPSeq-dep	51.04	54.88	52.89
Combination	53.07	57.41	**55.15**

4.3 Performance of the Inter-sentence CDR extraction

From Table 4 we can see that the performance of inter-sentence is quite low. The reason may be that:

- The inter-sentence level relations need more features and information to classify these implicit discourse relations. Only the raw word sequence may fail to capture some important information.
- It may be hard to learn the sequence representations between several sentences and the noise data also make confuse to the model.

Table 4. Performance of the Inter-sentence level methods.

Methods	***P*** **(%)**	***R*** **(%)**	***F*** **(%)**
Word	24.80	14.16	18.03
Word-position	33.49	13.79	**19.53**

4.4 Results of the CDR merging and post-processing

Then we merge the best intra-sentence level relations (**Combination**) and the best inter-sentence level relations (**Word-position**) to obtain the final CDR. The merging results are shown in Table 5. From the Table, we can see that adding inter-sentence level relation improves the F-score from 55.15% to 59.16%. After applying the post-processing rules to the system, the F-score achieves to 61.35%. In particular, the post-processing could help the system to pick up some missed CDR from the abstract and remove some false positives involving hypernym entities. As a supplement to the system, post-processing has a very strong effectiveness.

Table 5. Results of the CDR merging and post-processing.

System	***P*** **(%)**	***R*** **(%)**	***F*** **(%)**
Combination	53.07	57.41	55.15
CDR merging	60.19	58.16	59.16
+focused rules	55.48	66.41	60.46
+hypernym filter	58.38	64.63	**61.35**

4.5 Comparison with related work

Table 6 compares our system with the related work in the BioCreative V CDR task. All the systems are evaluated by the golden standard entities.

Table 6. Comparison with related work.

System	***P*** **(%)**	***R*** **(%)**	***F*** **(%)**
Xu *et al.*[4]	60.86/65.80*	53.10/68.57*	56.71/67.16*
Gu *et al.*[5]	62.00	55.10	58.30
Lowe *et al.*[3]	59.29	62.29	60.75
Zhou *et al.*[7]	55.56	68.39	61.31
Ours	58.38	64.63	**61.35**

For CDR extraction, Xu et al. [4] use large-scale prior knowledge database, Comparative Toxicogenomics Database (CTD), to extract the domain knowledge features. With the golden entities, they achieve the highest F-score 67.16% with CTD features (with the symbol '*') while the other result without CTD features. The features derived from the CTD provide the improvement from 56.71% to 67.16%. The knowledge databases play a critical role in CDR extraction as it could help extract the relations not exist in the training corpus effectively. Our system does not utilize large-scale knowledge bases, and could not achieve comparable performance using knowledge-based features in Xu et al. [4]. Recently, researchers have leveraged large-scale knowledge bases to learn knowledge representations, which show good performance for relation extraction [15]. We would like to leave the effect of knowledge representations as a problem for future work.

Gu et al. [5] use many lexical and dependency features with the maximum entropy classifiers. Compared with Gu et al. [5], our system does not need extensive feature engineering but achieves better performance. The reason may be that our CNN model could capture both sequence and dependency information more effectively. Lowe et al. [3] find CDR by a rule-based system and achieve 60.75% F-score. Their system is simple and effective. However, the handcrafted rules are hard to develop to a new dataset. Zhou et al. [7] integrate a feature-based model, a kernel-based and a neural network model into a uniform framework. Our system only uses the CNN, but achieve a slightly better results 61.35% F-score than their 61.31% F-score.

4.6 Error analysis

We perform an error analysis on the output of our final results (row 4 in Table 5) to detect the origins of false positives (FP) and false negatives (FN) errors, which are categorized in Fig. 5.

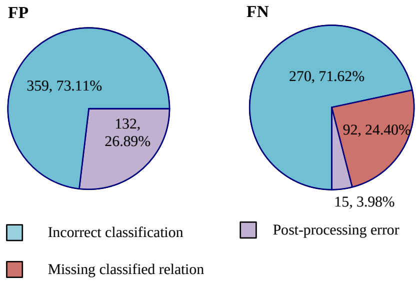

Fig. 5. Origins of FP and FN errors.

For FP in Fig. 5, two main error types are listed as follows:

- Incorrect classification: In spite of the detailed semantic representations, 73.11% FP come from the incorrect classification made by the intra- and inter- model. The main reason may be that sentence structure is complicated for both intra- and inter-level instances.
- Post-processing error: The focused rules bring 132 false CDR, with a proportion of 26.89%.

For FN in Fig. 5, three main error types are listed as follows:

- Incorrect classification: Among the 377 CDR that have not been extracted, 71.61% is caused by incorrect classification. Since it is difficult to find the relations spanning several sentences.
- Post-processing error: The hypernym filter removes 15 real CDR, with a proportion of 3.98%.
- Missing classified classification: 92 inter-sentence level instances are removed by the heuristic filtering rules in Section 3.3, which are not classified by our system at all. Because the sentence distance between the chemical and disease entities are more than 3.

5 Conclusion

Both semantic and syntactic information are effective for CDR extraction. Benefiting from the superior property of *k*-max pooling CNN, these information are well captured from word sequences and dependency structures for both intra- and inter-sentence level relation extraction. Furthermore, we propose weighted context representations for the sequence-based model to introduce external context of the two entities, which outperforms traditional context representations. Experiments on the BioCreative V CDR dataset show the effective of our sequence-based model, dependency-based model and their combination. In the future, we would like to encourage large-scale prior knowledge such as CTD and Wikipedia to improve extraction performance based on knowledge representation learning.

Acknowledgements. This research is supported by Natural Science Foundation of China (No. 61272375).

References

1. Dogan, R.I., Murray, G.C., Névéol, A., Lu, Z.Y.: Understanding PubMed user search behavior through log analysis. Database, doi: baw018. (2009).
2. Wei, C.H., Peng, Y.F., Leaman, Ret al.: Overview of the BioCreative V Chemical Disease Relation (CDR) Task. In: the fifth BioCreative Challenge Evaluation Workshop, pp. 154-166. (2015).
3. Lowe, D.M., O'Boyle, N.M and Sayle, R.A.: Efficient chemical-disease identification and relationship extraction using Wikipedia to improve recall. Database, doi: baw039. (2016).
4. Xu, J., Wu,Y.H., Zhang, Y.Y., Wang, J.Q., Lee and Xu, H.: CD-REST: a system for extracting chemical-induced disease relation in literature. Database, doi: baw036. (2016).
5. Gu, J.H., Qian, L.H and Zhou, G.D.: Chemical-induced disease relation extraction with various linguistic features. Database, doi: baw042. (2016).
6. Pons, E., Becker, B.F.H., Akhondi, S.A., Afzal, Z., van Mulligen, E.M and Kors, J.A.: Extraction of chemical-induced diseases using prior knowledge and textual information. Database, doi: baw046. (2016).
7. Zhou, H.W., Deng, H.J., Chen, L., Yang, Y.L., Jia, C. and Huang, D.G.: Exploiting syntactic and semantics information for chemical-disease relation extraction. Database, doi: baw048. (2016).
8. Gers, F.A and Schmidhuber, J.: Recurrent nets that time and count. In: Neural Networks: Como, 3: 189-194. (2000).
9. Nguyen, T.H and Grishman, R.: Relation extraction: perspective from convolutional neural networks. In: the NAACL Workshop on Vector Space Modeling for NLP, pp. 39-48. (2015).
10. Kalchbrenner, N., Grefenstette, R., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceeding of ACL, pp. 655-665. (2014).
11. Vu, N.T., Adel, H., Gupta, P., Schütze, H.: Combining recurrent and convolutional neural networks for relation classification. In: Proceedings of NAACL-HLT, pp. 534-539. (2016).
12. Zeng, D.J., Liu, K., Chen, Y.B., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of EMNLP, pp. 1753-1762. (2015).
13. Coletti, M.H., Bleich, H.L.: Medical subject headings used to search the biomedical literature. Journal of the American Medical Informatics Association, 8: 317-323. (2011).
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS, pp. 3111-3119. (2013).
15. Xie R.B, Liu Z.Y, Sun M. S.: Representation learning of knowledge graphs with hierarchical types. In: Proceedings of AAAI, pp. 2965-2971. (2016).