Li et al. BMC Medical Informatics and Decision Making 2018, 18(Suppl 2):60
https://doi.org/10.1186/s12911-018-0629-3

BMC Medical Informatics and Decision Making

RESEARCH Open Access

Chemical-induced disease extraction via recurrent piecewise convolutional neural networks

Haodi Li[1,2†], Ming Yang[3†], Qingcai Chen[1,2*], Buzhou Tang[1,2*], Xiaolong Wang[1,2] and Jun Yan[4]

From The 2nd International Workshop on Semantics-Powered Data Analytics
Kansas City, MO, USA. 13 November 2017

Abstract

Background: Extracting relationships between chemicals and diseases from unstructured literature have attracted plenty of attention since the relationships are very useful for a large number of biomedical applications such as drug repositioning and pharmacovigilance. A number of machine learning methods have been proposed for chemical-induced disease (CID) extraction due to some publicly available annotated corpora. Most of them suffer from time-consuming feature engineering except deep learning methods. In this paper, we propose a novel document-level deep learning method, called recurrent piecewise convolutional neural networks (RPCNN), for CID extraction.

Results: Experimental results on a benchmark dataset, the CDR (Chemical-induced Disease Relation) dataset of the BioCreative V challenge for CID extraction show that the highest precision, recall and F-score of our RPCNN-based CID extraction system are 65.24, 77.21 and 70.77%, which is competitive with other state-of-the-art systems.

Conclusions: A novel deep learning method is proposed for document-level CID extraction, where domain knowledge, piecewise strategy, attention mechanism, and multi-instance learning are combined together. The effectiveness of the method is proved by experiments conducted on a benchmark dataset.

Keywords: Chemical-induced disease, Relation extraction, Deep learning, Convolutional neural network

Background

Nowdays, there is more and more literature published with rich domain knowledge. The first step to reuse literature is to extract biomedical information from literature. Chemical-induced disease (CID), which refers to adverse drug reactions, is a type of important information, which can be used for drug safety monitoring and medicine development [1], has attracted more and more attentions.

During the last decade, there have been a large number of methods proposed for CID extraction [2], which can be classified into three categories: 1) statistics-based methods, 2) rule-based methods, and 3) machine learning-based methods. The statistics-based methods determine CIDs according to the distributions of chemicals and diseases. For example, Chen et al. [3] discovered drug side effects by analyzing co-occurrences of drugs and adverse reactions in biomedical literature. Mao et al. [4] used a similar method to mine drug side effects from social media. The limitation of statistic-based methods lies in their low precision, although they usually achieves high recall. Khoo et al. [5] used manually-constructed graphical patterns derived from syntactic parse trees to extract causal relations between drugs and adverse events in MEDLINE abstracts. The rule-based methods usually need domain experts, constructing rules is time-consuming, and the manually-crafted rules are not easily applicable to other corpora. To increase generalizability of rules, Xu and Wang [6] provided a method to learn syntactic patterns from sentences containing known drug side effect pairs for drug side effect extraction from biomedical literature. The manchine

* Correspondence: qingcai.chen@gmail.com; tangbuzhou@gmail.com
†Haodi Li and Ming Yang contributed equally to this work.
[1]Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology, Shenzhen, Guangdong, China
Full list of author information is available at the end of the article

learning-based methods are deployed for CID extaction due to some manually-annotated corpora, such as the corpus of the BioCreative V chemical-indcued disease relation (CDR) challenge [7] for CID extraction, are publically available. Support vector machine (SVM) is the most commonly used machine learning method. Xu et al. [8] won the BioCreative V CDR challenge using an SVM-based system. The feature engineering of the SVM-based system is terrible. To avoid fussy feature engineering, deep learning methods were applied to CID extraction [9], including convolutional neural networks (CNN) [10] and long short term memory neural networks (LSTM) [11]. In these systems, domain knowledge about adverse drug reactions, and some new techniques, such as piecewise strategy [12] and attention mechanism [13], widely used in other domains are not considered. Subsequently, Li et al. [14] adopted piecewise CNN to extract chemical-disease relations contained in intra-sentence and inter-sentence using a uniform model. Gu [15] improved the CNN model by adding syntactic information of cross-sentence, and the performance has been further improved. However, all these methods extract chemical-disease relations from single sentences or adjacent sentences. None of them consider document-level information. In a document, two entities usually do not appear only once, and it is difficult to determine which sentence or paragraph describes a relation or not. To facilitate efficient document-level relation extraction from biological text, Patrick [16] proposed Bi-affine Relation Attention Networks (BRAN), a combination of network architecture, multi-instance and multi-task learning. In this paper, we propose a novel document-level deep learning method for CID extraction, called recurrent piecewise convolutional neural networks (RPCNN). It should be noted that this paper is an extension of our previous paper [14].

Methods

Overview

There are usually two steps in chemical-induced disease extraction: 1) candidate generation – generating all possible related pairs of chemicals and diseases, denoted by <chemical, disease>; 2) candidate classification – determining whether each <chemical, disease> pair generated in the previous step is related.

Candidate generation

Given a biomedical record with m chemical mentions and n disease mentions, all $m \times n <$ chemical, disease> pairs can be recognized as candidates. In this study, we combine <chemical, disease> pairs that have the same chemical and disease identifiers together to form a candidate, denoted by <chemical identifier, disease identifier>. An example of candidate generation is shown in Table 1, where given a record with 2 chemical mentions (i.e., "terbutaline"×2) and 4 disease mentions (i.e., "Cardiovascular complications", "cardiovascular complications", "andpreterm labor"×2), as the two chemical mentions has the same MeSH (Medical Subject Headings) [17] identifier (i.e., D013726) and 4 disease mentions correspond to 2 MeSH identifiers (i.e, cardiovascular complications – D002318 and preterm labor – D007752), two candidates, that is, <D013726, D002318 > and < D013726, D007752>, are generated. Each candidate is a document-level candidate corresponding with multiple < chemical, disease> pairs, and each <chemical, disease> pair is an instance. Therefore, there are eight instances corresponding to two candidates in Table 1.

Candidate classification

A four-layer recurrent piecewise convolutional neural networks (RPCNN) is proposed for CID extraction as shown in Fig. 1, where piecewise CNN (the same as Li et al. [14]) is used to represent each instance of a candidate, and RNN is used to combine representations of each candidate's instances in a record together to obtain the document-level representation of the candidate.

Input layer

Given a candidate, the corresponding multiple instances I_0, I_1, …, I_m are arranged in descending order according to the length of context between the two entity mentions, which is measured by the number of words within the context. For each instance, we select the two entity mentions with context between them and context before or after them in the same sentence as the instance's input. To distinguish chemical entity mentions and disease mentions, "*<ENTC > ...</ENTC>*" and "*<ENTD> ... </ENTD>*", are further used to enclose them respectively. Then, an instance's input is divided into three parts: 1) $\boldsymbol{S}_{-1}$: context before the first entity mention (e.g., "Severe ... with" before "*<ENTC>* terbutaline *</ENTC>*" in Table 2); 2) $\boldsymbol{S}_0$: context between the two entity mentions (e.g., "for" in Table 2); and 3) $\boldsymbol{S}_1$: context after the second entity mention (e.g., "." after "*<ENTD>* preterm labor *</ENTD>*" in Table 2). Each word of an instance's input is represented by word embedding and embeddings of positions relative to chemcial and disease mentions (see Table 2). For convenience, the lengths of all instances' inputs (i.e., numbers of words within inputs) are set to the maximum (denoted by l). For instances with short input, paddings are appended to their input to make up the difference. Given an instance $<c, a>$ with input $S = w_1 w_2 \ldots w_l$, suppose that the positions of c and a in S are p_c and p_a respectively, word w_i can be represented by $x_i = [e_{w_i}^{\mathrm{T}}, e_{d_{ic}}^{\mathrm{T}}, e_{d_{ia}}^{\mathrm{T}}]^{\mathrm{T}}$, where $e_{w_i \in |V|}$, $e_{d_{ic}}$

Table 1 An example of candidate generation (Literature with chemical and disease mentions and their identifiers)

Position			Mention	Label		Identifier (MeSH)
start	end					
0	28		Cardiovascular complications	Disease		D002318
45	56		terbutaline	Chemical		D013726
71	84		preterm labor	Disease		D007752
93	121		cardiovascular complications	Disease		D002318
169	180		terbutaline	Chemical		D013726
185	198		preterm labor	Disease		D007752
Identifier (MeSH)	Chemical mention			Disease mention		
	position		mention	Position		mention
	start	end		start	end	
<D013726, D002318>	45	56	terbutaline	0	28	Cardiovascular complications
	45	56	terbutaline	93	121	Cardiovascular complications
	169	180	terbutaline	0	28	Cardiovascular complications
	169	180	terbutaline	93	121	Cardiovascular complications
	position			position		mention
	start	end		Start	End	
<D013726, D007752>	45	56		71	84	preterm labor
	45	56		185	198	preterm labor
	169	180		71	84	preterm labor
	169	180		185	198	preterm labor

Cardiovascular complications associated with terbutaline treatment for preterm labor
Abstract: Severe cardiovascular complications occurred in eight of 160 patients treated with terbutaline for preterm labor. Associated corticosteroid therapy and twin gestations appear to be predisposing factors. Potential mechanisms of the pathophysiology are briefly discussed

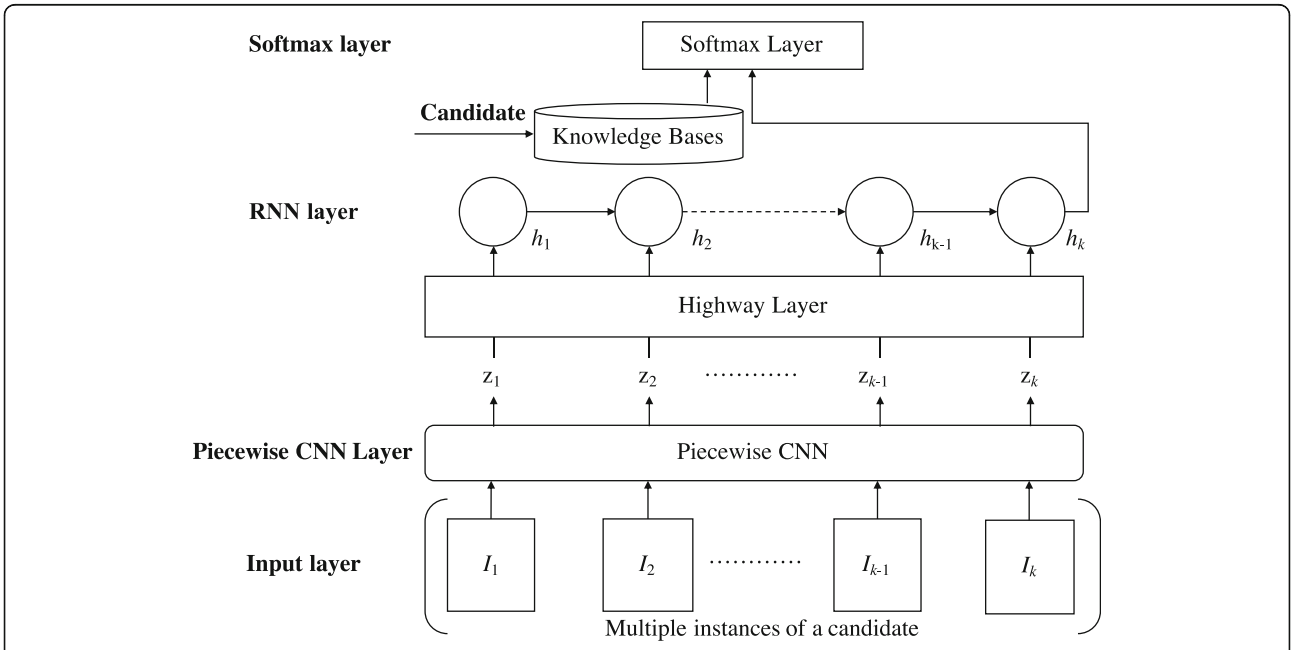

Fig. 1 Architecture of recurrent piecewise convolutional neural networks (RPCNN) for multi-instance learning

Table 2 Example of chemical position and disease position

Raw text:
Severe cardiovascular complications occurred in eight of 160 patients treated with terbutaline for preterm labor.
Input of a candidate:
Severe cardiovascular complications occurred in eight of 160 patients treated with <ENTC> terbutaline </ENTC> for <ENTD> preterm labor </ENTD>.
Chemical position and disease position (numbers in the first line under context are chemical positions and numbers in the second line under context are disease positions):
Severe cardiovascular complications occurred in eight of 160 patients treated with <ENTC>
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
-13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2
terbutaline </ENTC> for <ENTD> preterm labor </ENTD>.
0 1 1 2 2 2 2 3
-2 -2 -1 0 0 0 0 1

and $e_{d_{ia}}$ correspond to a d_w-dimensional word embedding, a d_{p^c}-dimensional position embedding and a d_{p^a}-dimensional position embedding, $d_{ic} = i - p_c$ and $d_{ia} = i - p_a$ are relative distances from w to c and a respectively ($-n + 1 \le d_{ic}, d_{ia} \le n - 1$), and $|V|$ is the word vocabulary. Then $S = w_1 w_2 w_3 ... w_l$ is represented by a matrix $x = [x_1, x_2, ..., x_l] \in R^{(d_w + d_{p^c} + d_{p^a}) \times l}$.

Piecewise convolutional layer

The convolutional layer takes the matrix of each instance' input x, and generates high-level feature vectors by convolving filters at multiple scales across x, where the filtes need to be learnt. Given a filter of size k, $t \in R^{(d_w + d_{p^c} + d_{p^a}) \times k}$, for example, feature vector $f = [f_1, f_2, ..., f_{l-k+1}]^{\mathrm{T}} \in R^{l-k+1}$ is generated by sliding filter t across S's input x with a convolution operator (take the rectified linear unit function (*Relu*) for example) as follows:

$$f_i = Relu(t \cdot x_{i:i+k-1} + b),$$

where $x_{i:i+k-1} = [x_i, x_{i+1}, ..., x_{i+k-1}]^{\mathrm{T}}$ is the context representation of $w_i w_{i+1} ... w_{i+k-1}$ within a k-word window, and $b \in R$ is a bias. Each filter corresponds to a high-level feature vector. Therefore, how many filters determines how many feature vectors we can obtain.

To reduce the spatial size of the representation of each instance, the number of parameters and computation, max pooling is adopted to select some important features from all the features generated in the convolutional layer:

$$\overline{f_t} = \max\left\{f_{t,1}, f_{t,2}, ..., f_{t,l+k-1}\right\},$$

where $(f_{t,1}, f_{t,2}, ..., f_{t,l+k-1})$ is the feature vector corresponding to filter t, and $\overline{f_t}$ is the maximum feature. If there are q filters, we a new q- dimensional vector is generated to represent S, denoted by $z = [\overline{f_1}, \overline{f_2}, ..., \overline{f_q}]^{\mathrm{T}}$. In addition, piecewise strategy that applies pooling to individual parts (i.e., S_{-1}, S_0 and S_1), and concatenates the outputs of all pooling layers is also adopted in our study.

Before pooling, attention mechanism is used to measure feature importances for each class as follows:

$$G_t = f_t{}^T \mathbf{M} \boldsymbol{W}^{\boldsymbol{classes}},$$

$$A_{i,j} = \frac{\exp(G_{i,j})}{\sum_{k=1}^{n} \exp(G_{k,j})},$$

where G is a correlation matrix between features f for each filter t and relation class embedding $\boldsymbol{W}^{\boldsymbol{classes}}$, $\mathbf{M}$ and $\boldsymbol{W}^{\boldsymbol{classes}}$ are weight matrix need to be learnt, A is an attention matrix, $A_{i,j}$ and $G_{i,j}$ are the (i, j)-th entry of A and G, respectively. We use a uniform distribution to initialize $\mathbf{M}$, and an identity matrix to initialize $\boldsymbol{W}^{\boldsymbol{classes}}$.

When the attention mechanism is adopted, the output of the pooling layer becomes:

$$\overline{f_{t,i}^{A}} = max_j (f_t A)_{i,j},$$

where $\overline{f_{t,i}^{A}}$ and $(f_t A)_{i,j}$ are the i-th item of $\overline{f_t^{A}}$ and the (i, j)-th item of $f_t A$, respectively.

RNN layer

In this layer, RNN is used to model multiple instances of a candidate. For each instance I_i, the corresponding RNN cell takes the output of the piecewise convolutional layer (i.e., z_i) and the previously hidden vector h_{i-1} as input, and output hidden vector h_i using a non-linear transformation function ρ, that is, $h_i = \rho(z_i, h_{i-1})$. The last hidden vector

h_m is used as the representation of multiple instances of a candidate, which is a document-level representation.

Softmax layer

In this layer, a fully connected neural network is used for classification. The neural network takes the following two parts as input: 1) h_m from the RNN layer presented above; 2) features extracted from four domain knowledge bases, the same as Xu et al.'s system [8], as follows:

(1) The CTD repository [18] that contains relationships between drugs and diseases, such as *inferred-association*, *therapeutic*, *marker/mechanism*, etc., manually summarized by experts.
(2) The Drugs and Indications Database (MEDI) [19] that records common drugs with common indications.
(3) SIDER (Drug Side Effects Database) [20] that records common drugs with common side effects.
(4) Medical Subject Headings (MeSH) that records superordinate and inferior structural relationships between drugs and the diseases.

The one-hot features extracted from domain knowledges are first converted into dense features (denoted by v) by a 1-layer neural network. For candidate classification, we use the sigmoid function as follows:

$$O(\boldsymbol{v}') = \left(1 + e^{\boldsymbol{u}\cdot\boldsymbol{v}'}\right)^{-1},$$

where $\boldsymbol{v}' = [\boldsymbol{h}_{\boldsymbol{m}}^{\mathbf{T}}, \boldsymbol{v}^{\mathrm{T}}]^{\mathrm{T}}$, and u is a weight vector.

Dataset

Our method is evaluated on the CDR corpus of the BioCreative V challenge. This corpus contains 1500 manually annotated PubMed record, 1000 out of 1500 records are used as training and development sets, and the remainder 500 records as test set. In the training and development sets, there are 10,550 chemical mentions, 8426 disease mentions, corresponding to 3829 and 2973 MeSH identifiers respectively. and 2050 relations. In the test set, there are 5385 chemical mentions, 4424 disease mentions, corresponding to 1988 and 1435 MeSH identifiers respectively, and 1066 relations.

Experimental settings

We start with a simple CNN-based system which only selects the last instance of every candidate in the input layer and does not use any one of domain knowledge, piecewise strategy or attention mechanism as baseline, and then compares it with CNN-based systems gradually using them and RPCNN. In addition, our best CNN-based and RPCNN-based systems are also compared with other state-of-the-art systems using a single machine learning method. Precision (P), recall (R) and F-score (F) are used to measure performance of all systems, which are calculated by the official evaluation tool of the BioCreative V organizer.

10-fold cross-validation is used to optimize all hyper-parameters of our system on the training and development sets. Finally, d_w, d_{p^c} and d_{p^a} are set to 30, 5 and 5 respectively. CBOW is deployed to initialize word embeddings on a large-scale unannotated corpus from Medline, and position embeddings are initialized by a uniform distribution. Filters at scales of 3 and 4 are selected and the numbers of filters are both set to 150. In the RNN layer, we used LSTM cell with 150 hidden states as the RNN cell. In the softmax layer, we follow Srivastava 's work [21] to randomly drop out units from networks to prevent overfitting during training, and set the dropout probability to 0.25. The number of units of the neural network for knowledge feature conversion is set to 120.

Results

The precision, recall and F-score of the baseline system (CNN in Table 3, where the best performance in each column is in bold) are 50.47, 55.61 and 52.92%. Similar with [8], the CNN-based systems is significantly improved by the domain knowledge. Take the baselien system as an example, when the domain knowledge is added, the system's F-score is improved by 15.72% (52.92% vs 68.64%). Both the piecewise strategy and attention mechanism are beneficial to the CNN-based systems and they are complementary to each other. For example, when the piecewise strategy is added into the baseline system (***CNN + piecewise*** in Table 3), the system's F-score increases from 52.92 to 54.20%, while when the attention mechanism is added to the baseline system before pooling (***CNN + attention***), the F-score slightly increases from 52.92 to 52.99%. When both the piecewise strategy and attention mechanism are together added to the baseline system (***CNN + attention + piecewise***), the system's F-score is further improved to 55.94%. When the domain knowledge is added, the effects of piecewise strategy and attention mechanism decrease. For example, the F-score difference between CNN using domain knowledge and ***CNN + piecewise*** using domain knowledge is 0.39%, while the F-score difference between corresponding systems without using domain knowledge is 1.28%. Among all CNN-based systems, the system that using domain knowledge, piecewise strategy and attention mechanism achieves highest F-score, which is 69.09%. The RPCNN-based system (***RPCNN***) outperforms ***CNN + attention + piecewise***. ***RPCNN*** without using domain knowledge achieves an F-score of 59.10%, higher than ***CNN + attention +***

Table 3 performance of our cnn-based and rpcnn-based systems for chemical-induced disease extraction

Methods	Without domain knowledge (%)			With domain knowledge (%)		
	P	*R*	*F*	*P*	*R*	*F*
CNN	50.47	55.61	52.92	63.70	74.40	68.64
CNN + piecewise	54.48	53.91	54.20	63.83	75.16	69.03
CNN + attention	48.40	58.54	52.99	62.28	76.58	68.69
CNN + attention+ piecewise	57.80	54.20	55.94	59.97	81.49	69.09
RPCNN	55.17	63.63	59.10	65.24	77.21	70.77

piecewise by 3.16%, while ***RPCNN*** using domain knowledge achieves an F-score of 70.77%, which is higher than that of ***CNN + attention + piecewise*** by 1.68%.

Moreover, our best CNN-based and RPCNN-based systems are also compared with other state-of-art systems using a single machine learning method, including Xu et al.'s system developed for the CDR task of the BioCreative V challenge [8], Zhou et al.'s LSTM-based and CNN-based systems [9], Gu et al.'s CNN-based system [15] and Patrick et al.'s BRAN-based system. Table 4 list the results of comparison, where "/" denotes no result report, and the best performance in each column is in bold. Compared with Xu et al.'s system, our RPCNN-based system achieves much higher F-score no matter whether the domain knowledge is used. The difference between the systems without using domain knowledge is 5.21% (55.94% vs 50.73%), while that between the systems using domain knowledge is 3.61% (70.77% vs 67.16%). Compared with Zhou et al.'s systems, our RPCNN-based system also achieves much higher F-score. The F-score difference between our RPCNN-based system and Zhou's systems arranges from 8.78 to 2.84%. Compared with Gu et al.'s system, though our CNN-based system does not perform better, our RPCNN-based system performs better by 1.90% in F-score. The Patrick et al.'s BRAN-based system achieves a higher F-score than our system by 3.00%, when it takes entity recogniton into account, which significantly improves the peformance of relation extraction. Without entity recognition multi-task objective, the BRAN-based's F-score is only 55.50%.

Discussion

In this paper, we propose RPCNN for CID extraction, where domain knowledge, piecewise strategy, attention mechanism and multi-instance learning are naturally combined. The RPCNN-based system on a benchmark corpus shows state-of-the-art performance.

Similar to previous studies on CNN-based relation extraction in other domains, the piecewise strategy and attention mechanism are effective in our CNN-based system. In our system, the attention mechanism makes it have the ability to handle some cases when the chemical mention is far away from the disease mention, especially they are not in one sentence. For example, a candidate < "AK", "cisplatin" > with the context of *"The primary outcome was acute kidney injury (<ENTD> AKI <ENTD>). RESULTS: We evaluated 143 patients who received single-agent <ENTC> cisplatin <ENTC>"*, where $S_1$is much longer and more complex than S_{-1} and S_0, is wrongly labeled as 0 when without using the piecewise strategy, but correctly labeled as 1 when using the piecewise strategy. However, tackling the two types of cases above mentioned are still challenging. We evaluate the performance of our system (CNN + attention+piecewise in Table 3) on tackling cases when the chemical mention and disease mention are not in one sentence. The precision, recall, and F-score are only 53.15, 26.07 and 34.99% respectively.

Compared with CNN-based systems, our RPCNN-based system performs better. The main reason is that RPCNN provides a document-level representation for every candidate as all corresponding instances are considered, while CNN only selects one instance to represent a candidate by removing other instances where there may be different descriptions about relations.

There may be two limitations of our study: 1) chemical mentions and disease mentions themselves are ignored in the input layer. The chemcial and disease mentions may be helpful for CID extraction. In the future work,

Table 4 Comparison between our systems and other state-of-the-art systems

Methods	Without domain knowledge (%)			With domain knowledge (%)		
	P	*R*	*F*	*P*	*R*	*F*
Xu et al. [8]	59.60	44.00	50.73	65.80	68.57	67.16
Zhou et al. (LSTM) [9]	54.91	51.41	53.10	/	/	/
Zhou et al. (CNN) [9]	41.13	55.25	47.16	/	/	/
Gu et al. (CNN) [15]	59.70	55.00	57.20	/	/	/
Patrick et al. (BRAN) [16]	55.60	70.80	62.10	/	/	/
Our CNN	57.80	54.20	55.94	59.97	81.49	69.09
Our RPCNN	55.17	63.63	59.10	65.24	77.21	70.77

we will have a try to integrate chemical and disease mentions in the input layer for further improvement. 2) The effectiveness of our method is validated on an independent test set from the same resource (BioCreative V challenge), but not on latest papers. We will manually label a corpus from PubMed including latest papers as another separate test set for further validation.

Conclusion

In this paper, we propose a novel document-level deep learning method for CID extraction. The proposed method naturally combines domain knowledge, piecewise strategy, attention mechanism and multi-instance learning together. The effectiveness of the method is validated on a benchmark corpus, and the system based on the proposed method shows competitive performance with other state-of-the-art systems.

Abbreviations

CDR: Chemical-induced Disease Relation; CID: Chemical-induced Disease; CNN: Convolutional Neural Network; LSTM: Long Short Term Memory Neural Networks; RNN: Recurrent Neural Network

Funding

This paper is supported in part by grants: National Natural Science Foundations of China (61573118, 61473101), Special Foundation for Technology Research Program of Guangdong Province (2015B010131010), Strategic Emerging Industry Development Special Funds of Shenzhen (JCYJ20160531192358466 and JCYJ20170307150528934) and Innovation Fund of Harbin Institute of Technology (HIT.NSRIF.2017052). This publication fee of this paper is supported by JCYJ20160531192358466. The funding agency was not involved in the design of this study, analysis and interpretation of data and the writing of the manuscript.

Availability of data and materials

The codes used in the experiments are now available at https://github.com/wglassly/CID_ATTCNN.

About this supplement

This article has been published as part of *BMC Medical Informatics and Decision Making* Volume 18 Supplement 2, 2018: Selected extended articles from the 2nd International Workshop on Semantics-Powered Data Analytics. The full contents of the supplement are available online at https://bmcmedinformdecismak.biomedcentral.com/articles/supplements/volume-18-supplement-2.

Authors' contributions

HL, MY, QC and BT designed the study together. HL and QC performed the experiments. HL, MY and BT analyzed the results, HL and BT write the manuscript. XW and JY reviewed and edited the manuscript. All authors read and approved the manuscript.

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology, Shenzhen, Guangdong, China. [2]Shenzhen Calligraphy Digital Simulation Technology Engineering Laboratory, Harbin Institute of Technology, Shenzhen, Guangdong, China. [3]Pharmacy Department, Shenzhen Second People's Hospital, First Affiliated Hospital of Shenzhen University, Guandong, Shenzhen, China. [4]Yidu Cloud (Beijing) Technology Co., Ltd, Beijing, China.

Published: 23 July 2018

References

1. Kang N, Singh B, Bui C, Afzal Z, van Mulligen EM, Kors JA. Knowledge-based extraction of adverse drug events from biomedical text. BMC Bioinformatics. 2014;15(1):64.
2. Zhou D, Zhong D, He Y. Biomedical relation extraction: from binary to complex. Comput Math Methods Med. 2014.
3. Chen ES, Hripcsak G, Xu H, Markatou M, Friedman C. Automated acquisition of disease–drug knowledge from biomedical and clinical documents: an initial study. J Am Med Inform Assoc. 2008;15(1):87–98.
4. Mao JJ, Chung A, Benton A, Hill S, Ungar L, Leonard CE, et al. Online discussion of drug side effects and discontinuation among breast cancer survivors. Pharmacoepidemiol Drug Saf. 2013;22(3):256–62.
5. Khoo CS, Chan S, Niu Y. Extracting causal knowledge from a medical database using graphical patterns. In: Proceedings of the 38th annual meeting on Association for Computational Linguistics. Association for Computational Linguistics; 2000. p. 336–43.
6. Xu R, Wang Q. Automatic construction of a large-scale and accurate drug-side-effect association knowledge base from biomedical literature. J Biomed Inform. 2014;51:191–9.
7. Li J, Sun Y, Johnson RJ, Sciaky D, Wei C-H, Leaman R, et al. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. Database. 2016;2016:baw068.
8. Xu J, Wu Y, Zhang Y, Wang J, Lee H-J, Xu H. CD-REST: a system for extracting chemical-induced disease relation in literature. Database. 2016; 2016:baw036.
9. Zhou H, Deng H, Chen L, Yang Y, Jia C, Huang D. Exploiting syntactic and semantics information for chemical–disease relation extraction. Database J Biol Databases Curation. 2016;
10. Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. Adv Neural Inf Proces Syst. 2015;1:649–57.
11. Liu P, Qiu X, Huang X.. Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101. 2016.
12. Zeng D, Liu K, Chen Y, Zhao J. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks, in Proceedings of EMNLP 2015, Lisbon, Portugal, September; 2015:17–21.
13. Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, et al. Attention-based bidirectional long short-term memory networks for relation classification. In: The 54th annual meeting of the Association for Computational Linguistics; 2016.
14. H. Li, Q. Chen, B. Tang and X. Wang. "Chemical-induced disease extraction via convolutional neural networks with attention," 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City, MO, USA, 2017. p. 1276–1279.
15. Gu et al. Chemical-induced disease relation extraction via convolutional neural network. Database (Oxford). 2017;2017:bax024.
16. Patrick Verga, Emma Strubell, Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL). 2018.
17. Lipscomb CE. Medical subject headings (MeSH). Bull Med Libr Assoc. 2000; 88(3):265.
18. Davis AP, Grondin CJ, Johnson RJ, Sciaky D, King BL, McMorran R, et al. The comparative toxicogenomics database: update 2017. Nucleic Acids Res. 2017;45(D1):D972–8.
19. Wei WQ, Cronin RM, H X, Lasko TA, Bastarache L, Denny JC. Development and evaluation of an ensemble resource linking medications to their indications. J Am Med Inform Assoc. 2013;20:954–61.
20. Kuhn M, Letunic I, Jensen LJ, Bork P. The SIDER database of drugs and side effects. Nucleic Acids Res. 2016;44(Database issue):D1075–9. https://doi.org/10.1093/nar/gkv1075.
21. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.