Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings

Dat Quoc Nguyen and **Karin Verspoor**
School of Computing and Information Systems
The University of Melbourne, Australia
{dqnguyen, karin.verspoor}@unimelb.edu.au

Abstract

We investigate the incorporation of character-based word representations into a standard CNN-based relation extraction model. We experiment with two common neural architectures, CNN and LSTM, to learn word vector representations from character embeddings. Through a task on the BioCreative-V CDR corpus, extracting relationships between chemicals and diseases, we show that models exploiting the character-based word representations improve on models that do not use this information, obtaining state-of-the-art result relative to previous neural approaches.

1 Introduction

Relation extraction, the task of extracting semantic relations between named entities mentioned in text, has become a key research topic in natural language processing (NLP) with a variety of practical applications (Bach and Badaskar, 2007). Traditional approaches for relation extraction are feature-based and kernel-based supervised learning approaches which utilize various lexical and syntactic features as well as knowledge base resources; see the comprehensive survey of these traditional approaches in Pawar et al. (2017). Recent research has shown that neural network (NN) models for relation extraction obtain state-of-the-art performance. Two major neural architectures for the task include the convolutional neural networks, CNNs, (Zeng et al., 2014; Nguyen and Grishman, 2015; Zeng et al., 2015; Lin et al., 2016; Jiang et al., 2016; Zeng et al., 2017; Huang and Wang, 2017) and long short-term memory networks, LSTMs (Miwa and Bansal, 2016; Zhang et al., 2017; Katiyar and Cardie, 2017; Ammar et al., 2017). We also find combinations of those two architectures (Nguyen and Grishman, 2016; Raj et al., 2017).

Relation extraction has attracted particular attention in the high-value biomedical domain. Scientific publications are the primary repository of biomedical knowledge, and given their increasing numbers, there is tremendous value in automating extraction of key discoveries (de Bruijn and Martin, 2002). Here, we focus on the task of understanding relations between chemicals and diseases, which has applications in many areas of biomedical research and healthcare including toxicology studies, drug discovery and drug safety surveillance (Wei et al., 2015). The importance of chemical-induced disease (CID) relation extraction is also evident from the fact that chemicals, diseases and their relations are among the most searched topics by PubMed users (Islamaj Dogan et al., 2009). In the CID relation extraction task formulation (Wei et al., 2015, 2016), CID relations are typically determined at document level, meaning that relations can be expressed across sentence boundaries; they can extend over distances of hundreds of word tokens. As LSTM models can be difficult to apply to very long word sequences (Bradbury et al., 2017), CNN models may be better suited for this task.

New domain-specific terms arise frequently in biomedical text data, requiring the capture of unknown words in practical relation extraction applications in this context. Recent research has shown that character-based word embeddings enable capture of unknown words, helping to improve performance on many NLP tasks (dos Santos and Gatti, 2014; Ma and Hovy, 2016; Lample et al., 2016; Plank et al., 2016; Nguyen et al., 2017). This may be particularly relevant for terms such as gene or chemical names, which often have identifiable morphological structure (Krallinger et al., 2017).

We investigate the value of character-based word embeddings in a standard CNN model for relation extraction (Zeng et al., 2014; Nguyen and Grishman, 2015). To the best of our knowledge,

there is no prior work addressing this.

We experiment with two common neural architectures of CNN and LSTM for learning the character-based embeddings, and evaluate the models on the benchmark BioCreative-V CDR corpus for chemical-induced disease relation extraction (Li et al., 2016a), obtaining state-of-the-art results.

2 Our modeling approach

This section describes our relation extraction models. They can be viewed as an extension of the well-known CNN model for relation extraction (Nguyen and Grishman, 2015), where we incorporate character-level representations of words.

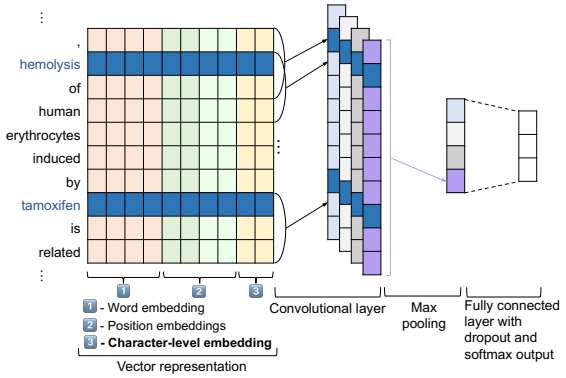

Figure 1: Our model architecture. Given the input relation mention marked with two entities "hemolysis" and "tamoxifen", the convolutional layer uses the window size $k = 3$ and the number of filters $m = 4$.

Figure 1 presents our model architecture. Given an input fixed-length sequence (i.e. a *relation mention*) of n word tokens $w_1, w_2, w_3, ..., w_n$,[1] marked with two entity mentions, the vector representation layer encodes each ith word in the input relation mention by a real-valued vector representation $\boldsymbol{v}_i \in \mathbb{R}^d$. The convolutional layer takes the input matrix $\boldsymbol{S} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n]^\mathsf{T}$ to extract high level features. These high level features are then fed into the max pooling layer to capture the most important features for generating a feature vector of the input relation mention. Finally, the feature vector is fed into a fully-connected neural network with softmax output to produce a probability distribution over relation types. For convenience, we detail the vector representation layer in Section 2.2 while the remaining layers appear in Section 2.1.

[1]We set n to be the length of the longest sequence and pad shorter sequences with a special "PAD" token.

2.1 CNN layers for relation extraction

Convolutional layer: This layer uses different filters to extract features from the input matrix $\boldsymbol{S} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n]^\mathsf{T} \in \mathbb{R}^{n \times d}$ by performing convolution operations. Given a window size k, a filter can be formalized as a weight matrix $\boldsymbol{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_k]^\mathsf{T} \in \mathbb{R}^{k \times d}$. For each filter $\boldsymbol{F}$, the convolution operation is performed to generate a feature map $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n-k+1}] \in \mathbb{R}^{n-k+1}$:

$$\boldsymbol{x}_j = g\big(\sum_{h=1}^{k} \boldsymbol{f}_h \boldsymbol{v}_{j+h-1} + b\big)$$

where $g(.)$ is some non-linear activation function and $b \in \mathbb{R}$ is a bias term.

Assume that we use m different weight matrix filters $\boldsymbol{F}^{(1)}$, $\boldsymbol{F}^{(2)}$, ..., $\boldsymbol{F}^{(m)} \in \mathbb{R}^{k \times d}$, the process above is then repeated m times, resulting in m feature maps $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(m)} \in \mathbb{R}^{n-k+1}$.

Max pooling layer: This layer aims to capture the most relevant features from each feature map $\boldsymbol{x}$ by applying the popular max-over-time pooling operation: $\hat{x} = \mathsf{max}\{\boldsymbol{x}\} = \mathsf{max}\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n-k+1}\}$. From m feature maps, the corresponding outputs are concatenated into a feature vector $\boldsymbol{z} = [\hat{x}^{(1)}, \hat{x}^{(2)}, ..., \hat{x}^{(m)}] \in \mathbb{R}^m$ to represent the input relation mention.

Softmax output: The feature vector $\boldsymbol{z}$ is then fed into a fully connected NN followed by a softmax layer for relation type classification. In addition, following Kim (2014), for regularization we apply dropout on $\boldsymbol{z}$ only during training. The softmax output procedure can be formalized as:

$$\boldsymbol{p} = \mathsf{softmax}\big(\mathbf{W}_1(\boldsymbol{z} * \boldsymbol{r}) + \boldsymbol{b}_1\big)$$

where $\boldsymbol{p} \in \mathbb{R}^t$ is the final output of the network in which t is the number of relation types, and $\mathbf{W}_1 \in \mathbb{R}^{t \times m}$ and $\boldsymbol{b}_1 \in \mathbb{R}^t$ are a transformation weight matrix and a bias vector, respectively. In addition, $*$ denotes an element-wise product and $\boldsymbol{r} \in \mathbb{R}^m$ is a vector of independent Bernoulli random variables, each with probability ρ of being 0 (Srivastava et al., 2014).

2.2 Input vector representation

This section presents the vector representation $\boldsymbol{v}_i \in \mathbb{R}^d$ for each ith word token in the input relation mention $w_1, w_2, w_3, ..., w_n$. Let word tokens w_{i_1} and w_{i_2} be two entity mentions in the input.[2] We obtain $\boldsymbol{v}_i$ by concatenating word embeddings $\boldsymbol{e}_{w_i} \in \mathbb{R}^{d_1}$, position embeddings $\boldsymbol{e}^{(p1)}_{i-i_1}$

[2]If an entity spans over multiple tokens, we take only the last token in the entity into account (Nguyen et al., 2016).

and $\boldsymbol{e}^{(p2)}_{i-i_2} \in \mathbb{R}^{d_2}$, and character-level embeddings $\boldsymbol{e}^{(c)}_{w_i} \in \mathbb{R}^{d_3}$ (so, $d = d_1 + 2 \times d_2 + d_3$):

$$\boldsymbol{v}_i = \boldsymbol{e}_{w_i} \circ \boldsymbol{e}^{(p1)}_{i-i_1} \circ \boldsymbol{e}^{(p2)}_{i-i_2} \circ \boldsymbol{e}^{(c)}_{w_i}$$

Word embeddings: Each word type w in the training data is represented by a real-valued word embedding $\boldsymbol{e}_w \in \mathbb{R}^{d_1}$.

Position embeddings: In relation extraction, we focus on assigning relation types to entity pairs. Words close to target entities are usually informative for identifying a relationship between them. Following Zeng et al. (2014), to specify entity pairs, we use position embeddings $\boldsymbol{e}^{(p1)}_{i-i_1}$ and $\boldsymbol{e}^{(p2)}_{i-i_2} \in \mathbb{R}^{d_2}$ to encode the relative distances $i - i_1$ and $i - i_2$ from each word w_i to entity mentions w_{i_1} and w_{i_2}, respectively.

Character-level embeddings: Given a word type w consisting of l characters $w = c_1 c_2 ... c_l$ where each jth character in w is represented by a character embedding $\boldsymbol{c}_j \in \mathbb{R}^{d_4}$, we investigate two approaches for learning character-based word embedding $\boldsymbol{e}^{(c)}_w \in \mathbb{R}^{d_3}$ from input $\boldsymbol{c}_{1:l} = [\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_l]^\mathsf{T}$ as follows:

(1) Using **CNN** (dos Santos and Gatti, 2014; Ma and Hovy, 2016): This CNN contains a convolutional layer to generate d_3 feature maps from the input $\boldsymbol{c}_{1:l}$, and a max pooling layer to produce a final vector $\boldsymbol{e}^{(c)}_w$ from those feature maps for representing the word w.

(2) Using a sequence BiLSTM (**BiLSTM$_{\text{seq}}$**) (Lample et al., 2016): In the BiLSTM$_{\text{seq}}$, the input is the sequence of l character embeddings $\boldsymbol{c}_{1:l}$, and the output is a concatenation of outputs of a forward LSTM (LSTM$_{\text{f}}$) reading the input in its regular order and a reverse LSTM (LSTM$_{\text{r}}$) reading the input in reverse:

$$\boldsymbol{e}^{(c)}_w = \text{BiLSTM}_{\text{seq}}(\boldsymbol{c}_{1:l}) \quad = \text{LSTM}_{\text{f}}(\boldsymbol{c}_{1:l}) \circ \text{LSTM}_{\text{r}}(\boldsymbol{c}_{l:1})$$

2.3 Model training

The baseline CNN model for relation extraction (Nguyen and Grishman, 2015) is denoted here as **CNN**. The extensions incorporating CNN and BiLSTM character-based word embeddings are **CNN+CNNchar** and **CNN+LSTMchar**, respectively. The model parameters, including word, position, and character embeddings, weight matrices and biases, are learned during training to minimize the model negative log likelihood (i.e. cross-entropy loss) with L_2 regularization.

3 Experiments

3.1 Experimental setup

We evaluate our models using the BC5CDR corpus (Li et al., 2016a) which is the benchmark dataset for the chemical-induced disease (CID) relation extraction task (Wei et al., 2015, 2016).[3] The corpus consists of 1500 PubMed abstracts: 500 for each of training, development and test. The training set is used to learn model parameters, the development set to select optimal hyper-parameters, and the test set to report final results. We make use of gold entity annotations in each case. For evaluation results, we measure the CID relation extraction performance with F1 score. More details of the dataset, evaluation protocol, and implementation are in the Appendix.

3.2 Main results

Table 1 compares the CID relation extraction results of our models to prior work. The first 11 rows report the performance of models that use the same experimental setup, without using additional training data or various features extracted from external knowledge base (KB) resources. The last 6 rows report results of models exploiting various kinds of features based on external relational KBs of chemicals and diseases, in which the last 4 SVM-based models are trained using both training and development sets.

The models exploiting more training data and external KB features obtained the best F1 scores. Panyam et al. (2016) and Xu et al. (2016) have shown that without KB features, their model performances (61.7% and 67.2%) are decreased by 5 and 11 points of F1 score, respectively.[4] Hence we find that external KB features are essential; we plan to extend our models to incorporate such KB features in future work.

In terms of models *not* exploiting external data or KB features (i.e. the first 11 rows in Table 1), our CNN+CNNchar and CNN+LSTMchar obtain the highest F1 scores; with 1+% absolute F1 improvements to the baseline CNN (p-value $<$ 0.05).[5] In addition, our models obtain 2+% higher

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

[4] Pons et al. (2016) and Peng et al. (2016) did not provide results without using the KB-based features. Xu et al. (2016) and Pons et al. (2016) did not provide results in using only the training set for learning models.

[5] Improvements are significant with p-value < 0.05 for a bootstrap significance test.

Model	P	R	F1
MaxEnt (Gu et al., 2016)	62.0	55.1	58.3
Pattern rule-based (Lowe et al., 2016)	59.3	62.3	60.8
LSTM-based (Zhou et al., 2016)	64.9	49.3	56.0
LSTM-based & PP (Zhou et al., 2016)	55.6	68.4	61.3
CNN-based (Gu et al., 2017)	60.9	59.5	60.2
CNN-based & PP (Gu et al., 2017)	55.7	68.1	61.3
BRAN (Verga et al., 2017)	55.6	70.8	**62.1**
SVM+APG (Panyam et al., 2018)	53.2	69.7	60.3
CNN	54.8	69.0	61.1
CNN+CNNchar	57.0	68.6	**62.3**
CNN+LSTMchar	56.8	68.8	62.2
Linear+TK (Panyam et al., 2016)	63.6	59.8	61.7
SVM (Peng et al., 2016)	62.1	64.2	63.1
SVM (+dev.) (Peng et al., 2016)	68.2	66.0	67.1
SVM (+dev.+18K) (Peng et al., 2016)	71.1	72.6	**71.8**
SVM (+dev.) (Xu et al., 2016)	65.8	68.6	67.2
SVM (+dev.) (Pons et al., 2016)	73.1	67.6	70.2

Table 1: Precision (P), Recall (R) and F1 scores (in %). "& PP" refers to the use of additional post-processing heuristic rules. "BRAN" denotes biaffine relation attention networks. "SVM+APG" denotes a model using SVM with All Path Graph kernel. "Linear+TK" denotes a model combining linear and tree kernel classifiers. "+dev." denotes the use of both training and development sets for learning models. Note that Peng et al. (2016) also used an extra training corpus of 18K weakly-annotated PubMed articles.

F1 score than the traditional feature-based models MaxEnt (Gu et al., 2016) and SVM+APG (Panyam et al., 2018). We also achieve 2+% higher F1 score than the LSTM- and CNN-based methods (Zhou et al., 2016; Gu et al., 2017) which exploit LSTM and CNN to learn relation mention representations from dependency tree-based paths.[6] Dependency trees have been actively used in traditional feature-based and kernel-based methods for relation extraction (Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; GuoDong et al., 2005; Mooney and Bunescu, 2006; Mintz et al., 2009) as well as in the biomedical domain (Fundel et al., 2007; Panyam et al., 2016, 2018; Quirk and Poon, 2017). Although we obtain better results, we believe dependency tree-based feature representations still have strong potential value. Note that to obtain dependency trees, previous work on CID relation extraction used the Stanford dependency parser (Chen and Manning, 2014). However, this dependency parser was trained on the Penn Treebank (in the newswire domain) (Marcus et al., 1993); training on a domain-specific treebank such as CRAFT (Bada et al., 2012) should help to improve results (Verspoor et al., 2012).

We also achieve slightly better scores than the more complex model BRAN (Verga et al., 2017), the Biaffine Relation Attention Network, based on the Transformer self-attention model (Vaswani et al., 2017). BRAN additionally uses byte pair encoding (Gage, 1994) to construct a vocabulary of subword units for tokenization. Using subword tokens to capture rare or unknown words has been demonstrated to be useful in machine translation (Sennrich et al., 2016) and likely captures similar information to character embeddings. However, Verga et al. (2017) do not provide comparative results using only original word tokens. Therefore, it is difficult to assess the usefulness specifically of using byte-pair encoded subword tokens in the CID relation extraction task, as compared to the impact of the full model architecture. We also plan to explore the usefulness of subword tokens in the baseline CNN for future work, to enable comparison with the improvement when using the character-based word embeddings.

It is worth noting that both CNN+CNNchar and CNN+LSTMchar return similar F1 scores, showing that in this case, using either CNN or BiLSTM to learn character-based word embeddings produces a similar improvement to the baseline. There does not appear to be any reason to prefer one of these in our relation extraction application.

4 Conclusion

In this paper, we have explored the value of integrating character-based word representations into a baseline CNN model for relation extraction. In particular, we investigate the use of two well-known neural architectures, CNN and LSTM, for learning character-based word representations. Experimental results on a benchmark chemical-disease relation extraction corpus show that the character-based representations help improve the baseline to attain state-of-the-art performance. Our models are suitable candidates to serve as future baselines for more complex models in the relation extraction task.

Acknowledgment: This work was supported by the ARC Discovery Project DP150101550.

[6] Zhou et al. (2016) and Gu et al. (2017) used the same post-processing heuristics to handle cases where models could not identify any CID relation between chemicals and diseases in an article, resulting in final F1 scores at 61.3%.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*. pages 265–283.

Waleed Ammar, Matthew Peters, Chandra Bhagavatula, and Russell Power. 2017. The AI2 system at SemEval-2017 Task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. pages 592–596.

Nguyen Bach and Sameer Badaskar. 2007. A Review of Relation Extraction. Technical report, Carnegie Mellon University.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, et al. 2012. Concept annotation in the CRAFT corpus. *BMC bioinformatics* 13(1):161.

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. Quasi-Recurrent Neural Networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Razvan Bunescu and Raymond Mooney. 2005. A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. pages 724–731.

Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. pages 740–750.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to Train good Word Embeddings for Biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. pages 166–174.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency Tree Kernels for Relation Extraction. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*. pages 423–429.

Berry de Bruijn and Joel Martin. 2002. Getting to the (c)ore of knowledge: mining biomedical literature. *International Journal of Medical Informatics* 67(1):7 – 18.

Cicero dos Santos and Maira Gatti. 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. pages 69–78.

Timothy Dozat. 2016. Incorporating Nesterov Momentum into Adam. In *Proceedings of the ICLR 2016 Workshop Track*.

Katrin Fundel, Robert Kffner, and Ralf Zimmer. 2007. RelExRelation extraction using dependency parse trees. *Bioinformatics* 23(3):365–371.

Philip Gage. 1994. A New Algorithm for Data Compression. *The C Users Journal* 12(2):23–38.

Jinghang Gu, Longhua Qian, and Guodong Zhou. 2016. Chemical-induced disease relation extraction with various linguistic features. *Database* 2016:baw042.

Jinghang Gu, Fuqing Sun, Longhua Qian, and Guodong Zhou. 2017. Chemical-induced disease relation extraction via convolutional neural network. *Database* 2017:bax024.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring Various Knowledge in Relation Extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. pages 427–434.

YiYao Huang and William Yang Wang. 2017. Deep Residual Learning for Weakly-Supervised Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1803–1807.

Rezarta Islamaj Dogan, G. Craig Murray, Aurlie Nvol, and Zhiyong Lu. 2009. Understanding PubMed user search behavior through log analysis. *Database* 2009.

Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. 2016. Relation Extraction with Multi-instance Multi-label Convolutional Neural Networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. pages 1471–1480.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint Extraction of Entity Mentions and Relations without Dependency Trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 917–928.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. pages 1746–1751.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *Transactions of the Association for Computational Linguistics* 4:313–327.

Martin Krallinger, Obdulia Rabal, Anlia Loureno, Julen Oyarzabal, and Alfonso Valencia. 2017. Information retrieval and text mining technologies for chemistry. *Chemical reviews* 117(12):7673–7761.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 260–270.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016a. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database* 2016:baw068.

Zhiheng Li, Zhihao Yang, Hongfei Lin, Jian Wang, Yingyi Gui, Yin Zhang, and Lei Wang. 2016b. CIDExtractor: A chemical-induced disease relation extraction system for biomedical literature. In *Proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine*. pages 994–1001.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 2124–2133.

Carolyn E. Lipscomb. 2000. Medical Subject Headings (MeSH). *Bulletin of the Medical Library Association* 88(3):265–266.

Daniel M. Lowe, Noel M. OBoyle, and Roger A. Sayle. 2016. Efficient chemical-disease identification and relationship extraction using Wikipedia to improve recall. *Database* 2016:baw039.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1064–1074.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* 19(2):313–330.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. pages 1003–1011.

Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1105–1116.

Raymond J. Mooney and Razvan C. Bunescu. 2006. Subsequence Kernels for Relation Extraction. In *Advances in Neural Information Processing Systems 18*, pages 171–178.

Dat Quoc Nguyen, Mark Dras, and Mark Johnson. 2017. A Novel Neural Network Model for Joint POS Tagging and Graph-based Dependency Parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. pages 134–142.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint Event Extraction via Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 300–309.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation Extraction: Perspective from Convolutional Neural Networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. pages 39–48.

Thien Huu Nguyen and Ralph Grishman. 2016. Combining Neural Networks and Log-linear Models to Improve Relation Extraction. In *Proceedings of IJCAI Workshop on Deep Learning for Artificial Intelligence*.

Nagesh C. Panyam, Karin Verspoor, Trevor Cohn, and Kotagiri Ramamohanarao. 2018. Exploiting graph kernels for high performance biomedical relation extraction. *Journal of Biomedical Semantics* 9(1):7.

Nagesh C. Panyam, Karin M. Verspoor, Trevor Cohn, and Kotagiri Ramamohanarao. 2016. Exploiting Tree Kernels for High Performance Chemical Induced Disease Relation Extraction. In *Proceedings of the 7th International Symposium on Semantic Mining in Biomedicine*. pages 42–47.

Sachin Pawar, Girish K. Palshikar, and Pushpak Bhattacharyya. 2017. Relation Extraction: A Survey. *arXiv preprint* arXiv:1712.05191.

Yifan Peng, Chih-Hsuan Wei, and Zhiyong Lu. 2016. Improving chemical disease relation extraction with rich features and weakly labeled data. *Journal of Cheminformatics* 8(1):53.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. pages 412–418.

Ewoud Pons, Benedikt F.H. Becker, Saber A. Akhondi, Zubair Afzal, Erik M. van Mulligen, and Jan A. Kors. 2016. Extraction of chemical-induced diseases using prior knowledge and textual information. *Database* 2016:baw046.

Chris Quirk and Hoifung Poon. 2017. Distant Supervision for Relation Extraction beyond the Sentence Boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. pages 1171–1182.

Desh Raj, Sunil Kumar Sahu, and Ashish Anand. 2017. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. In *Proceedings of the 21st Conference on Computational Natural Language Learning*. pages 311–321.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1715–1725.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15:1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Patrick Verga, Emma Strubell, Ofer Shai, and Andrew McCallum. 2017. Attending to All Mention Pairs for Full Abstract Biological Relation Extraction. In *Proceedings of the 6th Workshop on Automated Knowledge Base Construction*.

Karin Verspoor, Kevin Bretonnel Cohen, Arrick Lanfranchi, Colin Warner, Helen L Johnson, Christophe Roeder, Jinho D Choi, Christopher Funk, Yuriy Malenkiy, Miriam Eckert, et al. 2012. A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC bioinformatics* 13(1):207.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. 2015. Overview of the BioCreative V Chemical Disease Relation (CDR) Task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*. pages 154–166.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database* 2016:baw032.

Jun Xu, Yonghui Wu, Yaoyun Zhang, Jingqi Wang, Hee-Jin Lee, and Hua Xu. 2016. CD-REST: a system for extracting chemical-induced disease relation in literature. *Database* 2016:baw036.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. pages 2335–2344.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating Relation Paths in Neural Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1768–1777.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-End Neural Relation Extraction with Global Optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1730–1740.

Huiwei Zhou, Huijie Deng, Long Chen, Yunlong Yang, Chen Jia, and Degen Huang. 2016. Exploiting syntactic and semantics information for chemicaldisease relation extraction. *Database* 2016:baw048.

Appendix

Dataset and evaluation protocol: We evaluate our models using the BC5CDR corpus (Li et al., 2016a), which is the benchmark dataset for the BioCreative-V shared task on chemical-induced disease (CID) relation extraction (Wei et al., 2015, 2016).[7] The BC5CDR corpus consists of 1500 PubMed abstracts: 500 each for training, development and test set. In all articles, chemical and disease entities were manually annotated using the Medical Subject Headings (MeSH) concept identifiers (Lipscomb, 2000).

CID relations were manually annotated for each relevant pair of chemical and disease concept identifiers at the *document level* rather than for each pair of entity mentions (i.e. the relation annotations are not tied to specific mention annotations). Figure 2 shows examples of CID relations. We follow Gu et al. (2016) (see relation instance construction and hypernym filtering sections) and Gu

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

1601297|t|Electrocardiographic evidence of myocardial injury in psychiatrically hospitalized cocaine abusers.
1601297|a|The electrocardiograms (ECG) of 99 cocaine-abusing patients were compared with the ECGs of 50 schizophrenic controls. Eleven of the cocaine abusers and none of the controls had ECG evidence of significant myocardial injury defined as myocardial infarction, ischemia, and bundle branch block.

1601297	33	50	myocardial injury	Disease	D009202
1601297	83	90	cocaine	Chemical	D003042
1601297	135	142	cocaine	Chemical	D003042
1601297	194	207	schizophrenic	Disease	D012559
1601297	232	239	cocaine	Chemical	D003042
1601297	305	322	myocardial injury	Disease	D009202
1601297	334	355	myocardial infarction	Disease	D009203
1601297	357	365	ischemia	Disease	D007511
1601297	371	390	bundle branch block	Disease	D002037
1601297	CID	D003042	D009203		
1601297	CID	D003042	D002037		

Figure 2: A part of an annotated PubMed article.

et al. (2017) to transfer these annotations to *mention level* relation annotations.

In the evaluation phase, mention-level classification decisions must be transferred to the document level. Following Gu et al. (2016), Li et al. (2016b) and Gu et al. (2017), these are derived from either (i) a pair of entity mentions that has been positively classified to form a CID relation based on the document or (ii) a pair of entity mentions that co-occurs in the document, and that has been annotated as having a CID relation in a document in the training set.

In an article, a pair of chemical and disease concept identifiers may have multiple entity mention pairs, expressed in different relation mentions.

The longest relation mention has about 400 word tokens; the longest word has 37 characters.

We use the training set to learn model parameters, the development set to select optimal hyperparameters, and the test to report final results using gold entity annotations. For evaluation results, we measure the CID relation extraction performance using F1 score.

Implementation details: We implement CNN, CNN+CNNchar, CNN+LSTMchar using Keras (Chollet et al., 2015) with a TensorFlow backend (Abadi et al., 2016), and use a fixed random seed. For both CNN+CNNchar and CNN+LSTMchar, character embeddings are randomly initialized with 25 dimensions, i.e. $d_4 = 25$. For CNNchar, the window size is 5 and the number of filters at 50, resulting in $d_3 = 50$. For LSTMchar, we set the number of LSTM units at 25, also resulting in $d_3 = 50$.

For all three models, position embeddings are randomly initialized with 50 dimensions, i.e. $d_2 = 50$. Word embeddings are initialized by using 200-dimensional pre-trained word vectors from Chiu et al. (2016), i.e. $d_1 = 200$; and word types (including a special "UNK" word token representing unknown words), which are not in the embedding list, are initialized randomly. Following Kiperwasser and Goldberg (2016), the "UNK" word embedding is learned during training by replacing each word token w appearing n_w times in the training set with "UNK" with probability $\mathsf{p}_{unk}(w) = \frac{0.25}{0.25+n_w}$ (this procedure only involves the word embedding part in the input vector representation layer). We use ReLU for the activation function g, and fix the window size k at 5 and the L_2 regularization value at 0.001.

We train the models with Stochastic gradient descent using Nadam (Dozat, 2016). For training, we run for 50 epochs. We perform a grid search to select the optimal hyperparameters by monitoring the F1 score after each training epoch on the development set. Here, we select the initial Nadam learning rate $\lambda \in \{5\text{e-}06, 1\text{e-}05, 5\text{e-}05, 1\text{e-}04, 5\text{e-}04\}$, the number of filters $m \in \{100, 200, 300, 400, 500\}$ and the dropout probability $\rho \in \{0.25, 0.5\}$. We choose the model with highest F1 on the development set, which is then applied to the test set for the evaluation phase.