The Impact of Self-Interaction Attention on the Extraction of Drug-Drug Interactions

Luca Putelli[1,2], **Alfonso E. Gerevini**[1], **Alberto Lavelli**[2], **Ivan Serina**[1]
[1]Università degli Studi di Brescia, [2]Fondazione Bruno Kessler
{alfonso.gerevini, ivan.serina}@unibs.it, {l.putelli, lavelli}@fbk.eu

Abstract

Since a large amount of medical treatments requires the assumption of multiple drugs, the discovery of how these interact with each other, potentially causing health problems to the patients, is the subject of a huge quantity of documents. In order to obtain this information from free text, several methods involving deep learning have been proposed over the years. In this paper we introduce a Recurrent Neural Network-based method combined with the Self-Interaction Attention Mechanism. Such a method is applied to the DDI2013-Extraction task, a popular challenge concerning the extraction and the classification of drug-drug interactions. Our focus is to show its effect over the tendency to predict the majority class and how it differs from the other types of attention mechanisms.

1 Introduction

Given the increase of publications regarding side effects, adverse drug reactions and, more in general, how the assumption of drugs can cause risks of health issues that may affect patients, a large quantity of free-text containing crucial information has become available. For doctors and researchers, accessing this information is a very demanding task, given the number and the complexity of such documents.

Hence, the automatic extraction of Drug-Drug Interactions (DDI), i.e. situations where the simultaneous assumption of drugs can cause adverse drug reactions, is the goal of the DDIExtraction-2013 task (Segura-Bedmar et al., 2014). DDIs have to be extracted from a corpus of free-text sentences, combining machine learning with natural language processing (NLP).

Starting from the introduction of word embedding techniques like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) for word representation, Recurrent Neural Networks (RNN) and in particular Long Short Term Memory networks (LSTM) have become the state-of-the-art technology for most of natural language processing tasks like text classification or relation extraction.

The main idea behind the attention mechanism (Bahdanau et al., 2014) is that the model “pays attention" only to the parts of the input where the most relevant information is present. In our case, this mechanism assigns a higher weight to the most influential words, i.e. the ones which describe an interaction between drugs.

Several attention mechanisms have been proposed in the last few years (Hu, 2018), in particular self-interaction mechanism (Zheng et al., 2018) applies attention with a different weight vector for each word in the sequence, producing a matrix that represents the influence between all word pairs. We consider this information very meaningful, especially in a task like this one where we need to discover connections between pairs of words.

In this paper we show how self-interaction attention improves the results in the DDI-2013 task, comparing it to other types of attention mechanisms. Given that this dataset is strongly unbalanced, the main focus of the analysis is how each attention mechanism deals with the tendency to predict the majority class.

2 Related work

The best performing teams in the DDI-2013 original challenge (Segura-Bedmar et al., 2014) used SVM (Björne et al., 2013) but, more recently, Convolutional Neural Networks (CNN) (Liu et al., 2016), (Quan et al., 2016) and mostly Recurrent Neural Networks (RNN) have proved to be the new state of the art.

Kumar and Anand (2017) propose a double LSTM. The sentences are processed by two different bidirectional LSTM layers: one followed by a max-pooling layer and the other one by a custom made attention-pooling layer that assign weights to words. Furthermore Zhang et al. (2018) design a multi-path LSTM neural network. Three parallel bidirectional LSTM layers process the sentence sequence and a fourth one processes the shortest dependency path between the two candidate drugs in the dependency tree. The output of these four layers is merged and handled by another bidirectional LSTM layer.

Zheng et al. (2017) apply attention directly to word vectors, creating a “candidate-drugs-oriented" input which is processed by a single LSTM layer.

Yi et al. (2017) use a RNN with Gated Recurrent Units (GRU) (Cho et al., 2014) instead of LSTM units, followed by a standard attention mechanism, and exploits information contained in other sentences with a custom made sentence attention mechanism.

Putelli et al. (2019) introduce an LSTM model followed by a self-interaction attention mechanism which computes, for each pair of words, a vector representing how much it is related to the other. These vectors are concatenated into a single one which is passed to a classification layer. In this paper, starting from the results reported in Putelli et al. (2019), we improve the input representation, the negative filtering and extend the analysis of self-interaction attention, comparing it to more standard attention mechanisms.

3 Dataset description

This dataset was released for the shared challenge SemEval 2013 - Task 9 (Segura-Bedmar et al., 2014) and contains annotated documents from the biomedical literature. In particular, there are two different sources: abstracts from MEDLINE research articles and texts from DrugBank.

Every document is divided into sentences and, for each sentence, the dataset provides annotations of every drug mentioned. The task requires to classify all the possible $\binom{n}{2}$ pairs of n drugs mentioned in the given sentences. The dataset provides the instances with their classification value.

There are five different classes: **unrelated**: there is no relation between the two drugs mentioned; **effect**: the text describes the effect of the drug-drug interaction; **advise**: the text recommends to avoid the simultaneous assumption of two drugs; **mechanism**: the text describes an anomaly of the absorption of a drug, if assumed simultaneously with another one; **int**: the text states a generic interaction between the drugs.

4 Pre-processing

The pre-processing phase exploits the “en_core_web_sm" model of spaCy[1], a Python tool for Natural Language Processing, and it is composed by these steps:

Substitution: after tokenization and POS-tagging, the drug mention tokens are replaced by the standard terms `PairDrug1` and `PairDrug2`. In the particular case when the pair is composed by two mentions of the same drug, these are replaced by `NoPair`. Every other drug mentioned in the sentence is replaced with the generic name `Drug`.

Shortest dependency path: spaCy produces the dependency tree associated to the sentence, with tokens as nodes and dependency relations between the words as edges. Then, we calculate the shortest path in the dependency tree between `PairDrug1` and `PairDrug2`.

Offset features: given a word w in the sentence, D_1 is calculated as the distance (in terms of words) from the first drug mention, divided by the length of the sentence. Similarly, D_2 is calculated as the distance from the second drug mention.

4.1 Negative instance filtering

The DDI-2013 dataset contains many “negative instances", i.e. instances that belong to the unrelated class. In an unbalanced dataset, machine learning algorithms are more likely to classify a new instance over the majority class, leading to poor performance for the minority classes (Weiss and Provost, 2001). Given that previous studies (Chowdhury and Lavelli, 2013; Kumar and Anand, 2017; Zheng et al., 2017) have demonstrated a positive effect of reducing the number of negative instances on this dataset, we have filtered out some instances from the training-set relying only on the structure of the sentence, starting from the pairs of drugs with the same name. In

[1] https://spacy.io

addition to this case, we can filter out a candidate pair if the two drug mentions appear in coordinate structure, checking the shortest dependency path between the two drug mentions.If they are not connected by a path, i.e. there is no grammatical relation between them, the candidate pair is filtered out.

While other works like (Kumar and Anand, 2017) and (Liu et al., 2016) apply custom-made rules for this dataset (such as regular expressions), our choice is to keep the pre-processing phase as general as possible, defining an approach that can be applied for other relation extraction tasks.

5 Model description

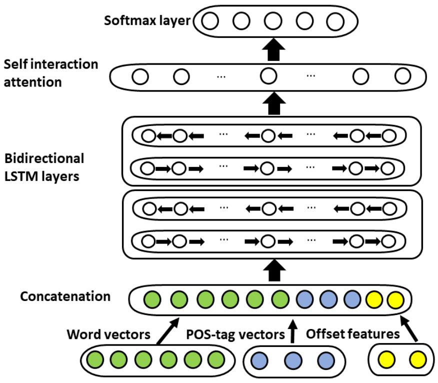

Figure 1: Model architecture

In this section we present the LSTM-based model (Figure 1), the self-attention mechanism and how it is used for relation extraction.

5.1 Embedding

Each word in our corpus is represented with a vector of length 200. These vectors are obtained with a Word2Vec (Mikolov et al., 2013) fine-tuning. We initialized a Word2Vec model with the vectors obtained by the authors of McDonald et al. (2018) the same algorithm over PubMed abstracts and PMC texts, and trained our Word2Vec model using the DDI-2013 corpus.

PoS tags are represented with vectors of length 4. These are obtained applying the Word2Vec method to the sequence of PoS tags in our corpus.

5.2 Bidirectional LSTM layer

A Recurrent neural network is a deep learning model for processing sequential data, like natural language sentences. Its issues with vanishing gradient are avoided using LSTM cells (Hochreiter and Schmidhuber, 1997; Gers et al., 2000), which allow to process longer and more complex sequences. Given $x_1, x_2 \dots x_m$, h_{t-1} and c_{t-1} where m is the length of the sentence and $x_i \in \mathbb{R}^d$ is the vector obtained by concatenating the embedded features, h_{t-1} and c_{t-1} are the hidden state and the cell state of the previous LSTM cell (h_0 and c_0 are initialized as zero vectors), new hidden state and cell state values are computed as follows:

$$\begin{aligned}
\hat{c}_t &= \tanh(W_c[h_{t_i}, x_t] + b_c) \\
i_t &= \sigma(W_i[h_{t_i}, x_t] + b_i) \\
f_t &= \sigma(W_f[h_{t_i}, x_t] + b_f) \\
o_t &= \sigma(W_o[h_{t_i}, x_t] + b_o) \\
c_t &= i_t * \hat{c}_t + f_t * c_{t-1} \\
h_t &= \tanh(c_t) * o_t
\end{aligned}$$

with σ being the sigmoid activation function and $*$ denoting the element wise product. W_f, W_i, W_o, $W_c \in \mathbb{R}^{(N+d)\times N}$ are weight matrices and b_f, b_i, b_o, $b_c \in \mathbb{R}^N$ are bias vectors. Weight matrices and bias vectors are randomly initialized and learned by the neural network during the training phase. N is the LSTM layer size and d is the dimension of the feature vector for each input word. The vectors in square brackets are concatenated.

Bidirectional LSTM not only computes the input sequence in the order of the sentence but also backwards (Schuster and Paliwal, 1997). Hence, we can compute h^r using the same equations described earlier but reversing the word sequence. Given h_t computed in the sentence order and h^r_t in the reversed order, the output of the t bidirectional LSTM cell h^b_t is the result of the concatenation of h_t and h^r_t.

5.3 Sentence representation and attention mechanisms

The LSTM layers produce, for each word input w_i, a vector $h_i \in \mathbb{R}^n$ which is the result of computing every word from the start of the sentence to w_i. Hence, given a sentence of length m, h_m can be considered as the sentence representation produced by the LSTM layer. So, for a sentence classification task, h_m can be used as the input to a fully connected layer that provides the classification.

Even if they perform better than simple RNNs, LSTM neural networks have difficulties preserving dependencies between distant words (Raffel and Ellis, 2015) and, especially for long sentences, h_m may not be influenced by the first words or may be affected by less relevant words. The **Attention** mechanism (Bahdanau et al., 2014; Kadlec et al., 2016) deals with these problems taking into consideration each h_i, computing weights α_i for each word contribution:

$$u_i = \tanh(W_a h_i + b_a)$$
$$\alpha_i = softmax(u_i) = exp(u_i)/\sum_{k=1}^{n} exp(u_k)$$

where $W_a \in \mathbb{R}^{N \times N}$ and $b_a \in \mathbb{R}^{N}$.

The attention mechanism outputs the *sentence representation*

$$s = \sum_{i=1}^{m} \alpha_i h_i$$

The **Context Attention** mechanism (Yang et al., 2016) is more complex. In order to enhance the importance of the words for the meaning of the sentence, this uses a *word level context vector* u_w of additional weights for the calculation of α_i:

$$\alpha_i = softmax(u_w^T u_i)$$

As proposed by Zheng et al. (2018), **Self-Interaction Attention** mechanism uses multiple v_i for each word w_i instead of using a single one. This way, we can extract the influence (called *action*) between the *action controller* w_i and the rest of the sentence, i.e. each w_k for $k \in \{1, m\}$. The action of w_i is calculated as follows:

$$s_i = \sum_{k=1}^{m} \alpha_{i,k} u_i$$
$$\alpha_{i_k} = exp(v_k^T u_i)/\sum_{j=1}^{m} exp(v_j^T u_i)$$

with u_i defined in the same way as the standard attention mechanism.

5.4 Model architecture

In order to obtain also in this case a *context vector* representing the sentence, in Zheng et al. (2018) each s_i is aggregated into a single vector s as its average, maximum or even applying another standard attention layer. In our model we choose to avoid any pooling operations and to concatenate instead each s_i, creating a *flattened representation* (Du et al., 2018) and passing it to the classification layer.

The model designed (see Figure 1) and tested for the DDI-2013 Relation Extraction task includes the following layers: three parallel **embedding layers**: one with pre-trained word vectors, one with pre-trained PoS tag vectors and one that calculates the embedding of the offset features; two **bidirectional LSTM layers** that process the word sequence; the **self-interaction attention mechanism**; a **fully connected layer** with 5 neurons (one for each class) and `softmax` activation function that provides the classification.

In our experiments, we compare this model with similar configurations obtained substituting the self-interaction attention with the standard attention layer introduced by Bahdanau et al. (2014) and the context-attention of Yang et al. (2016).

6 Results and discussion

Our models are implemented using Keras library with Tensorflow backend. We perform a simple random hyper-parameter search (Bergstra and Bengio, 2012) in order to optimize the learning phase and avoiding overfitting, using a subset of sentences as validation set.

6.1 Evaluation

We have tested our two models with different input configurations: using only word vectors, using word and PoS tag vectors or adding also offset features.

In Table 1 we show the recall measure for each input configuration. The effect of self-interaction is also verified through the Friedman test (Friedman, 1937): for all input configurations, the model with self-interaction attention performs better than the other configurations with a level of confidence equals to 99%. Similarly, the simple Attention Mechanism obtains better performances with respect to the Context Attention with confidence of 99% (see Figure 2).

In Table 2 we show the F-Score for each class of the dataset. The overall performance of the configuration including word vectors, PoS tagging and offset features as input is considered also in Table 3.

In Table 3 we compare our results with other state-of-the-art methods and compare the overall performance of the three attention mechanisms. The Context-Att obtains results similar to those of most of the approaches based on Convolution Neural Networks and worse than most of LSTM-based models.

In terms of F-Score, Word Attention LSTM (Zheng et al., 2017) outperforms our approach and the other LSTM-based models by more than 4%. As we discussed in (Putelli et al., 2019), we have tried to replicate their model but we could not obtain the same results. Furthermore, their attention mechanism aimed to creating a “candidate-drugs-oriented" input did not improve the performance.

Input	No Attention	Context-Att	Attention	Self-Int-Att
Word	64.44	65.32	66.60	**69.72**
Word+Tag	65.37	65.20	67.57	**68.95**
Word+Tag+Offset	60.67	65.82	69.47	**70.88**

Table 1: Overall recall (%) comparison with different attention mechanisms and input configurations. For each input configuration, the best recall is marked in bold.

	Effect				Mechanism			
Input	No Att	C-Att	Att	Self-Int	No Att	C-Att	Att	Self-Int
Word	0.68	0.71	**0.72**	0.70	0.69	0.72	0.72	0.70
Word+Tag	0.67	0.70	0.70	0.69	0.71	0.73	0.74	0.70
Word+Tag+Offset	0.65	0.70	0.70	0.69	0.68	0.73	0.74	**0.76**
	Advise				Int			
Input	No Att	C-Att	Att	Self-Int	No Att	C-Att	Att	Self-Int
Word	0.77	0.71	0.74	0.78	0.53	0.49	0.45	0.45
Word+Tag	0.78	0.73	0.77	0.77	**0.55**	0.50	0.45	0.43
Word+Tag+Offset	0.74	0.75	**0.79**	0.78	0.50	0.52	0.50	0.49

Table 2: Detailed F-Score comparison with different configurations and attention mechanisms. For each class, the best F-Score is marked in bold.

Method	P(%)	R(%)	F(%)
UTurku (SVM)	73.2	49.9	59.4
FBK-irst (SVM)	64.6	65.6	65.1
Zhao SCNN	72.5	65.1	68.6
Liu CNN	75.7	64.7	69.8
Multi-Channel	76.0	65.3	70.2
Context-Att	**75.9**	**65.8**	**70.5**
Joint-LSTMs	73.4	69.7	71.5
Self-Int	**73.0**	**70.9**	**71.9**
GRU	73.7	70.8	72.2
Attention	**75.6**	**69.5**	**72.4**
SDP-LSTM	74.1	71.8	72.9
Word-Att LSTM	78.4	76.2	77.3

Table 3: Comparison with overall precision (P), recall (R) and F-Score (F) of other state-of-the-art methods: , ordered by F. Our models are marked in bold, results higher than ours are marked in red.

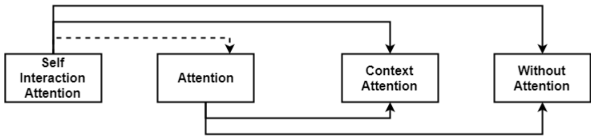

Figure 2: Recall comparison for models with different attention mechanisms for Word+Tag+Offset. The continue arrow means 99% confidence, while the dashed arrow means 95%.

7 Conclusions and future work

We have compared the self-interaction attention model to alternative configurations using the standard attention mechanism introduced by Bahdanau et al. (2014) and the context-attention mechanism of Yang et al. (2016).

Our experiments show that the self-interaction mechanism improves the performance with respect to other versions, in particular reducing the tendency of predicting the majority class, hence decreasing the number of false negatives. The standard attention mechanism produces better results than the context attention.

As future work, our objective is to exploit or adapt the Transformer architecture (Vaswani et al., 2017), which has become quite popular for machine translation tasks and relies almost only on attention mechanisms, and apply it to relation extraction tasks like DDI-2013.

Another direction includes the exploitation of a different pre-trained language modeling. For example, BioBERT (Lee et al., 2019) obtains good results for several NLP tasks like Named Entity Recognition or Question Answering and we plan to apply it to our task.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305, February.

Jari Björne, Suwisa Kaewphan, and Tapio Salakoski. 2013. UTurku: Drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 651–659, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Md. Faisal Mahbub Chowdhury and Alberto Lavelli. 2013. FBK-irst : A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, pages 351–355.

Jinhua Du, Jingguang Han, Andy Way, and Dadong Wan. 2018. Multi-level structured self-attentions for distantly supervised relation extraction. *CoRR*, abs/1809.00699.

Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12:2451–2471.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80, 12.

Dichao Hu. 2018. An introductory survey on attention mechanisms in NLP problems. *CoRR*, abs/1811.05544.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. *CoRR*, abs/1603.01547.

Sunil Kumar and Ashish Anand. 2017. Drug-drug interaction extraction from biomedical text using long short term memory network. *CoRR*, abs/1701.08303.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2016. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, 2016.

Ryan McDonald, Georgios-Ioannis Brokos, and Ion Androutsopoulos. 2018. Deep relevance ranking using enhanced document-query interactions. *CoRR*, abs/1809.01682.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Luca Putelli, Alfonso E. Gerevini, Alberto Lavelli, and Ivan Serina. 2019. Applying self-interaction attention for extracting drug-drug interactions. In *Proceedings of 18th International Conference of the Italian Association for Artificial Intelligence*.

Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. 2016. Multichannel convolutional neural network for biological relation extraction. *BioMed research international*, 2016.

Colin Raffel and Daniel P. W. Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *CoRR*, abs/1512.08756.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2014. Lessons learnt from the DDIExtraction-2013 shared task. *Journal of Biomedical Informatics*, 51:152–164.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Gary Weiss and Foster Provost. 2001. The effect of class distribution on classifier learning: An empirical study. Technical report, Department of Computer Science, Rutgers University.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*.

Zibo Yi, Shasha Li, Jie Yu, Yusong Tan, Qingbo Wu, Hong Yuan, and Ting Wang. 2017. Drug-drug interaction extraction via recurrent neural network with multiple attention layers. In *International Conference on Advanced Data Mining and Applications*, pages 554–566. Springer.

Yijia Zhang, Wei Zheng, Hongfei Lin, Jian Wang, Zhihao Yang, and Michel Dumontier. 2018. Drug-drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths. *Bioinformatics*, 34(5):828–835.

Wei Zheng, Hongfei Lin, Ling Luo, Zhehuan Zhao, Zhengguang Li, Zhang Yijia, Zhihao Yang, and Jian Wang. 2017. An attention-based effective neural model for drug-drug interactions extraction. *BMC Bioinformatics*, 18, 12.

Jianming Zheng, Fei Cai, Taihua Shao, and Honghui Chen. 2018. Self-interaction attention mechanism-based text representation for document classification. *Applied Sciences*, 8(4).