Tree-LSTM and Cross-Corpus Training for Extracting Biomedical Relationships from Text

Joël Legrand[1], Yannick Toussaint[1], Chedy Raïssi[1], and Adrien Coulet[1]

LORIA (CNRS, Inria Nancy-Grand Est, University of Lorraine)
Campus Scientifique, Nancy, France
joel.legrand@loria.fr

Abstract. A key aspect of machine learning-based relationship extraction algorithms is the availability of training data. Manually annotated corpora are valuable resources for this task, but the time and expertise required for their development explain that still few corpora are available. For tasks related to precision medicine, most of them are rather small (*i.e.,* hundreds of sentences) or focus on specialized relationships (*e.g.*, drug-drug interactions) that rarely fit what one wants to extract. In this paper, we experiment Tree-LSTM, to extract relationships from biomedical texts with high performance. In addition we show that relatively large corpora, even when focusing on a distinct type of relationships, can be use to improve the performance of deep learning-based system for relationship extraction tasks for which initial resources are scarce.

Keywords: Deep learning, NLP, Tree-LSTM, Cross-corpus training, relationship extraction, pharmacogenomic relationships

1 Introduction

With the exponential growth of biomedical publications, it becomes difficult for researchers and clinicians to keep themselves up-to-date. In precision medicine, which takes into account individual variability in genes, environment, and lifestyle for each patient, this problem is even more acute, since it involves many research areas (such as pharmacogenomics, toxocogenomics and environment impact studies), thus many more publications. Relationship Extraction (RE) from the literature has been proposed to extract automatically structured information from unstructured text. In particular, deep learning methods have demonstrated good ability for such task [29]. One drawback of these methods is that they generally require a large amount of annotated data in order to obtain reasonable performances. As a first step towards RE for precision medicine, we study the extraction of pharmacogenomic relationship, *i.e.*, relationships between individual gene variations and drug response phenotypes. But to our knowledge, no adequate corpus exists with these specific relationships annotated. Building one may be particularly time consuming and expensive since it involves complex entities and requires trained annotators. However, several large and small corpora have been manually annotated with other type of biomedical relationships and made available. Because these corpora share the same language (*i.e.*, English) and thus a common syntax, one might wonder if these resources developed for slightly different task may be reused for extracting relationships for which resources are scarce, *i.e.*, only few hundred of annotated sentences are available.

Beside deep learning methods mentioned above, many approaches for RE have been proposed. Co-occurrence-based methods assumes that two entities mentioned frequently in the same unit of text (such as a sentence or a paragraph) are related [10]. Rule-base methods use manually designed, or learned, rules consisting of word morphosyntactic features or sentence-level syntactic features [8, 9]. These methods have the advantage of requiring few or no annotated data. Also, rules can often be interpreted, since it is possible to associate a meaning to a rule. If the set of rules has been learned, these interpretations can help experts in better understanding a phenomena. In contrast, machine learning methods learn statistical models by training over annotated corpora and then recognize relationships in unannotated text [12]. These methods tend to achieve better performances than the formers.

Within machine learning methods, deep learning ones are used to model complex structures such as natural language and successfully applied to various Natural Language Processing (NLP) tasks. In particular, it as been applied to RE with success from annotated corpora . While other methods mainly depend on the quality of extracted features derived from preexisting NLP systems, deep learning models automatically learn lexical features using continuous word vector representations and sentence level features using deep neural network such as Convolutional Neural Network (CNN) [7] or Recursive Neural Networks (RNN) [22]. Despite their advantages, these models strongly depend on the existence of large training corpora, which make them difficult to use for tasks associated with scarce resources.

In this paper we propose, within a specific RE task for which only few training data are available, to investigate how large annotated corpora may be used to improve performances of deep neural networks. For that purpose, we used the tree-structured Long Short Term Memory (Tree-LSTM) model [24], for which variants have proven to be suitable and effective for RE [20]. We experimented using two relatively small corpora (SNPPhenA and EU-ADR) along with two larger corpora (SemEval 2013 DDI and ADE). Section 2 presents various deep learning methods used for RE while Section 3 details the Tree-LSTM used in our experiments. Section 4 describes the set of corpus used in this study and Section 5 presents our experiments and results.

2 Related work

Deep learning models, based on continuous word representations, have been proposed to overcome the problem of sparsity inherent to NLP [15]. In Collobert et al. [7], the authors proposed an unified CNN architecture to tackle various NLP problems traditionally handle with statistical approaches. They obtained state-of-the-art performances for several tasks, while avoiding the hand design of task specific features. These results led to progress on NLP topics such as machine translation [6], question-answering [3] and RE.

In particular, Zeng et al. [29] showed that CNN models can also be applied to the task of RE. They built such a model to learn a sentence representation, given word and word position embeddings. This representation is then used to feed a softmax classifier [1]. To improve the performance of the RE, other authors consider elements of syntax within the embedding provided to the model: Xu et al. [26] use the path of grammatical dependencies between two entities, which is provided by a dependency parsing; Yang et al. [27] include the relative positions of words in a dependency tree. They also take dependency based context (child and parent nodes) into account during the convolution.

While these CNN models incorporate syntactic knowledge in their embeddings, other approaches go further by proposing neural networks which topology is adapt-

ing to the syntactic structure of the sentence. In particular, Recursive Neural Networks (RNN) have been proposed to adapt to tree structures resulting from constituency parsing [17, 22]. In that vein, Tai et al. [24] introduced a Tree-LSTM, a generalization of LSTM for tree-structured network topologies, which allows to process trees with arbitrary branching factors.

The first model to make use of RNN for a RE task was proposed by Liu et al. [19]. The authors introduced a CNN-based model applied on the shortest dependency path, augmented with a RNN-based feature designed to model subtrees attached to the shortest path. Miwa and Bansal [20] introduced a variant of the Tree-LSTM used to compute bidirectional (bottom-up and top-down) tree representations that perform relationship classification. Their model use different weight matrices depending on whether a node belong to the shortest path or not.

For the extraction of biomedical relationships, CNN have been successively applied by adapting the Multichannel Variable-Size CNN [28] to the extraction of drug-drug interactions (DDI) [18] and protein-protein interactions [21].

3 Model

In this paper we used the Tree-LSTM model described in Tai et al. [24] and more specifically its Child-Sum version. This model is suitable for processing dependency trees since it handles trees with arbitrary branching factors and no order between children of a node. The model computes a score for each possible type or relationship (*e.g.* negative, positive or speculative) between two identified entities. The number of possible relationship types depends on the task (see Section 4). In this section, we first introduce the features used in this study; then, we detail the Tree-LSTM model; finally, we present the scoring layer.

3.1 Input layer

Our Tree-LSTM network is fed with *word embeddings* (*i.e.*, continuous vectors) of dimension d_w. Extra *entity embeddings* of size d_e are used to distinguish the two entities involved in a relation from the other words. The first entity is mapped to the embedding *first entity*, the second to the embedding *second entity* and all the other words to the embedding *other*. Word and entity embeddings are concatenated to form the input of the Tree-LSTM.

3.2 Tree-LSTM

The Tree-LSTM model processes the dependency tree associated with an input sentence in a bottom-up manner. This is done by recursively processing the nodes of the tree, using their child representations as input. The transition function for a node j and a set of children $C(j)$ is given by the following set of equations:

$$\begin{aligned}
\tilde{h}_t &= \sum_{k \in C(j)} h_k \\
i_j &= \sigma(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}) \\
f_{jk} &= \sigma(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}) \\
o_j &= \sigma(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)})
\end{aligned}$$

$$u_j = \tanh(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)})$$
$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k$$
$$h_j = o_j \odot \tanh(c_j)$$

, where σ denotes the logistic function, $\odot$ the element-wise multiplication, $x_j \in \mathcal{R}^{d_w + d_e}$ is the input for node j, $h_k \in \mathcal{R}^{d_h}$ is the hidden state of the k^{th} child. Each Tree-LSTM unit is a collection of vectors: an input gate i_j, a forget gate f_{jk}, an output gate o_j, a memory cell c_j and and hidden state h_j. The matrices W and U and the vectors b are the weight and bias parameters to train.

3.3 Scoring layer

The Tree-LSTM layer processes any dependency tree recursively from its leaves to its parent nodes to lastly provide a representation for the root node, *i.e.*, the node that spans all the other nodes. This representation takes the entire sentence into account and is used to feed a single layer neural network classifier, which outputs a score for each possible class. These scores are interpreted as probabilities using a softmax classifier [1].

4 Datasets

We explore how RE tasks that focus on a type of biomedical relationships associated with scarce resources may take advantage from additional resources, in other words how completing a small training corpus with a larger one may help the RE task, even if the latter is annotated with a different type of relationship. For this purpose, we selected two relatively small corpora, SNPPhenA and the EU-ADR corpus, and two larger, the SemEval 2013 DDI corpus and the ADE corpus. All these corpora are publicly available and focus on relationships between two of the three entities of interest in pharmacogenomics (*i.e.* gene or gene variations, drugs and phenotypes). Table 4.2 summarizes the main characteristics of these four corpora and the following section details them.

4.1 Small corpora

- **SNPPhenA** [2] is a corpus of abstracts of biomedical publications, obtained from PubMed[1], annotated with two type of entities: *single nucleotide polymorphisms* (SNPs) and *phenotypes*. Relationships between these entities are also annotated and classified in 3 categories: *positive*, *negative* and *neutral* relationships. *neutral* relationships is used when no relationship is mentioned in the sentence between the two entities.
- **EU-ADR** [25] is a corpus of abstracts obtained from PubMed and annotated with *drugs*, *disorders* and targets (*proteins/genes* or *gene variants*) entities. The corpus is composed of 3 subcorpora, focusing on target-disease, target-drug and drug-disease relationships. Each of them consist of 100 abstracts. Relationships are classified in 3 categories: *positive*, *speculative* and *negative associations* (PA, SA and NA respectively). In [4], performances are assessed over the `TRUE` class,

[1] https://www.ncbi.nlm.nih.gov/pubmed/

which is composed of the classes PA, SA and NA, in contrast with the `FALSE` class made with sentences where the two entities co-occur, with no relationship between them.

4.2 Large corpora

- **SemEval 2013 DDI** [14] consists of texts from DrugBank and MEDLINE and is annotated with drugs. Drug mentions are categorized in several types: *drug*, *brand*, *group* and *drug_n* (*i.e.*, active substances not approved for human use). Relationships between two drug mentions are annotated and classified in 4 categories: *mechanism*, *effect*, *advice* and *int*. *int* is the broader and default category for DDI, when no more detail can be provided.
- **ADE-EXT** (Adverse Drug Effect corpus, extended) [13] consists of MEDLINE case reports, annotated with *drug* and *conditions* (*e.g.*, diseases, signs and symptoms) along with relationships between them, when one is mentioned.

Corpus	**Subcorpus**	**Train Size**		**Test Size**		**#Entity**	**#Relation**
		sent.	rel.	sent.	rel.	**Types**	**Types**
SNPPhenA	–	362	935	121	365	2	3
EU-ADR	drug-disease	244	176	–	–	4	3
	drug-target	247	310			4	3
	target-disease	355	262			4	3
SemEval 2013 DDI	DrugBank	5,675	3,805	973	889	4	4
	MEDLINE	1,301	232	326	95	4	4
ADE-EXT	–	5,939	6,701	–	–	2	1

Table 1. Main characteristics of the corpora. Two corpora are divided in subcorpora. The sizes of the training and test corpora are reported in term of number of sentences (sent.) and relationships (rel.). EU-ADR and ADR-EXT have no proper test corpus.

5 Experiments

5.1 Training

Following Collobert et al. [7], our LSTM network was trained by minimizing a log-likelihood function over the training data. All parameters, including weight, biases and embeddings were updated via Backpropagation Through Structure (BPTS) [11].

5.2 Experimental Settings

Dependency trees were obtained using the Stanford Parser [5]. Hyper-parameters were tuned using 10 fold cross-validation and fixed to $d_w = 100$, $d_e = 10$, $d_h = 200$. We applied dropout regularization [23] after every Tree-LSTM unit and after the embedding layers. The drop probability for each connexion was fixed to 0.25. Word

embeddings were pre-trained using the method described in Lebret and Collobert [16] and trained on PubMed abstracts.

We evaluated performances in terms of precision (P), recall (R) and f-measure (F). For multi-label classification, we report the macro-average performance. Because no proper test corpus is provided with EU-ADR, we performed a 10 fold cross-validation using 10% of the corpus for the validation and 10% for the test of our models. For SNPPhenA, we performed a cross-validation using 10% of the corpus for the validation and the provided test corpus for testing. Each result is an average of 5 experiments starting from different random weight initialization.

5.3 Cross-corpus study

In this subsection, we present our cross-corpus training strategy and its results. For the cross-corpus experiments, the same network was used for the different corpora (*i.e.*, same embedding layer and Tree-LSTM weights), except for the scorer, which is different for each corpus as the number and types of relationship may change. During the training phase, we randomly pick training sentences from the mixed corpora. Table 2 presents the results of the cross-corpus study. We observe that using additional data consistently improved the performances and interestingly, this phenomenon occurs even for corpora with different types of entities such as the combination of SNPPhenA and SemEval 2013 DDI.

Test Corpus	Train corpus	P	R	F	σ_F
SNPPhenA	SNPPhenA	58.9	73.8	65.5	0.041
	+ SemEval 2013 DDI	65.2	71.1	**68.0**	0.047
	+ ADE-EXT	62.8	72.1	67.2	0.034
EU-ADR drug-disease	EU-ADR drug-disease	74.8	84.1	79.1	0.123
	+ SemEval 2013 DDI	74.8	90.6	**82.0**	0.131
	+ ADE-EXT	73.9	88.2	80.4	0.137
EU-ADR drug-target	EU-ADR drug-target	72.4	90.6	80.2	0.109
	+ SemEval 2013 DDI	71.9	95.5	**82.5**	0.085
	+ ADE-EXT	70.2	96.7	80.9	0.092
EU-ADR target-disease	EU-ADR target-disease	77.0	89.7	82.7	0.064
	+ SemEval 2013 DDI	77.4	91.6	**83.9**	0.082
	+ ADE-EXT	77.7	89.5	83.3	0.069

Table 2. Impact of cross-corpus training in terms of precision (P), recall (R) and f1-score (F). σ_F is the standard deviation of the f1-score.

5.4 Comparison with the state of the art

Table 3 presents a comparison of performances obtained with our approach *versus* two state-of-the-art systems applied to the RE task associated respectively with SNPPhenA and EU-ADR, reported in Bokharaeian et al. [2] and Bravo et al. [4]. Our results for the SNPPhenA corpus are obtained using, for each fold, an ensemble of 5 models starting from different random initialization, following Legrand and Collobert [17]. We report the 10 folds average performance. For the EU-ADR experiments, we did not observed any improvement using our ensemble of models. Both state-of-the-art systems use a combination of a shallow linguistic kernel with a kernel that exploits deep syntactic features. Our approach outperforms

the performances reported for SNPPhenA and for the drug-disease subcorpus or EU-ADR. For the two other EU-ADR subcorpora, our approach achieves similar performances, while relying on few automatically extracted features whereas the two baselines use several tuned orthographic and morpho-syntactic features.

Test corpus	Train corpus	P	R	F
	Bokharaeian et al. [2]			
	SNPPhenA	56.6	59.8	58.2
SNPPhenA	**This work**			
	SNPPhenA ensemble + ADE-EXT	64.5	75.2	**69.4**
	Bravo et al. [4]			
EU-ADR	EU-ADR drug-disease	70.2	93.2	79.3
drug-disease	**This work**			
	EU-ADR drug-disease + SemEval 2013 DDI	74.8	90.6	**82.0**
	Bravo et al. [4]			
EU-ADR	EU-ADR drug-target	74.2	97.4	**83.0**
drug-target	**This work**			
	EU-ADR drug-target + SemEval 2013 DDI	71.9	95.5	82.5
	Bravo et al. [4]			
EU-ADR	EU-ADR target-disease	75.1	97.7	**84.6**
target-disease	**This work**			
	EU-ADR target-disease + SemEval 2013 DDI	77.4	91.6	83.9

Table 3. Performance comparison with the state of the art in terms of precision (P), recall (R) and f1-score (F).

6 Conclusion

In this paper, we empirically demonstrated that corpora developed for specific RE tasks may be use to improve the performance of deep learning-based system for other RE tasks, for which initial resources are scarce. In addition, our rather simple approach led to state-of-the-art results with the SNPPhenA corpus and to near state-of-the-art results with the EU-ADR corpora. This study is an interesting first step for the extraction of pharmacogenomic relationships for which no appropriate corpus is available. Surprisingly, the best results where consistently obtained using the SemEval 2013 DDI corpus as additional data, even for RE tasks that doesn't involve drugs like EU-ADR target-disease. Likewise, one might have thought that the ADE-EXT corpus could have been more asuitable for the EU-ADR drug-disease corpus, since it shares common entities. Several ideas should be explored to better understand this phenomenon, such as the differences of relation and entity types between the different corpora as well as the differences of textual source (*e.g.* medical case report for ADE-EXT, research articles for the others).

References

1. Christopher M Bishop. *Pattern recognition and machine learning*. 2006.
2. Behrouz Bokharaeian, Alberto Diaz, et al. Snpphena: a corpus for extracting ranked associations of single-nucleotide polymorphisms and phenotypes from literature. *Journal of Biomedical Semantics*, 2017.

3. Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.
4. Àlex Bravo, Janet Piñero, et al. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC Bioinformatics*, 2015.
5. Danqi Chen and Christopher D. Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, 2014.
6. Kyunghyun Cho, Bart Van Merriënboer, et al. On the properties of neural machine translation: Encoder-decoder approaches. 2014.
7. Ronan Collobert, Jason Weston, et al. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2011.
8. Adrien Coulet, Nigam H Shah, Yael Garten, Mark Musen, and Russ B Altman. Using text to build semantic networks for pharmacogenomics. *Journal of Biomedical Informatics*, 2010.
9. Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relexrelation extraction using dependency parse trees. *Bioinformatics*, 2007.
10. Yael Garten and Russ B. Altman. Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text. *BMC Bioinfo.*, 2009.
11. Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of ICNN*, 1996.
12. Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, 2005.
13. Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. Extraction of potential adverse drug events from medical case reports. *Journal of Biomedical Semantics*, 2012.
14. María Herrero-Zazo, Isabel Segura-Bedmar, et al. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics*, 2013.
15. Fei Huang and Alexander Yates. Distributional representations for handling sparsity in supervised sequence-labeling. In *Proceedings of ACL*, 2009.
16. Rémi Lebret and Ronan Collobert. Word emdeddings through hellinger pca. *EACL*, 2013.
17. Joël Legrand and Ronan Collobert. Joint RNN-based greedy parsing and word composition. *arXiv preprint arXiv:1412.7028*, 2014.
18. Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Drug-drug interaction extraction via convolutional neural networks. *CMMM*, 2016.
19. Yang Liu, Furu Wei, et al. A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646*, 2015.
20. Makoto Miwa and Mohit Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.
21. Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. Multichannel convolutional neural network for biological relation extraction. *BioMed research international*, 2016.
22. Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing with compositional vector grammars. In *Proceedings of ACL*, 2013.
23. Nitish Srivastava, Geoffrey E Hinton, et al. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.
24. Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured LSTM networks. In *ACL*, 2015.
25. Erik M Van Mulligen, Annie Fourrier-Reglat, et al. The EU-ADR corpus: annotated drugs, diseases, targets, and their relationships. *Journal of Biomedical Informatics*, 2012.
26. Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *EMNLP*, 2015.
27. Yunlun Yang, Yunhai Tong, et al. A position encoding convolutional neural network based on dependency tree for relation classification. In *EMNLP*, 2016.
28. Wenpeng Yin and Hinrich Schütze. Multichannel variable-size convolution for sentence classification. *arXiv preprint arXiv:1603.04513*, 2016.
29. Daojian Zeng, Kang Liu, Siwei Lai, et al. Relation classification via convolutional deep neural network. In *proceedings of COLING*, 2014.