CROSS-CORPUS TRAINING WITH TREELSTM FOR THE EXTRACTION OF BIOMEDICAL RELATIONSHIPS FROM TEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

A bottleneck problem in machine learning-based relationship extraction (RE) algorithms, and particularly of deep learning-based ones, is the availability of training data in the form of annotated corpora. For specific domains, such as biomedicine, the long time and high expertise required for the development of manually annotated corpora explain that most of the existing one are relatively small (*i.e.,* hundreds of sentences). Beside, larger corpora focusing on general or domain-specific relationships (such as citizenship or drug-drug interactions) have been developed. In this paper, we study how large annotated corpora developed for alternative tasks may improve the performances on biomedicine related tasks, for which few annotated resources are available. We experiment two deep learning-based models to extract relationships from biomedical texts with high performance. The first one combine locally extracted features using a Convolutional Neural Network (CNN) model, while the second exploit the syntactic structure of sentences using a Recursive Neural Network (RNN) architecture. Our experiments show that, contrary to the former, the latter benefits from a cross-corpus learning strategy to improve the performance of relationship extraction tasks. Indeed our approach leads to state-of-the-art performances for four biomedical RE tasks for which few annotated resources are available (less than 400 manually annotated sentences). This may be particularly impactful in specialized domains in which training resources are scarce, because they would benefit from the training data of other domains for which large annotated corpora does exist.

1 INTRODUCTION

Relationship Extraction (RE) from text is a Natural Language Processing (NLP) task that aims at extracting automatically and summarizing in a structured form the unstructured information of texts. Deep learning methods have demonstrated good ability for such tasks Zeng et al. (2014), but one of their drawbacks is that they generally require a large amount of training data, *i.e.*, text corpora where entities and relationships between them are annotated, in order to obtain reasonable performances. The building of such a corpus for a specific task, such as those of interest in biomedicine, is time consuming and expensive because it implies complex entities (*e.g.*, genomic variations, complex symptoms), complex relationships (which may be hypothetical, contextualized, negated, n-ary) and requires trained annotators. This explain why only few and relatively small (*i.e.*, few hundreds of sentences) corpora are available, making these resources particularly valuable. Among these tasks, one can mention the extraction of genomic variations-phenotype relationships for which only a manually annotated corpus of 362 sentences, named SNPPhena exists (Bokharaeian et al., 2017). Beside, several larger corpora have been manually annotated with biomedical or general-domain relationships and made available (Hachey et al., 2012; Herrero-Zazo et al., 2013; Gurulingappa et al., 2012). Because these corpora share the same language (*i.e.*, English) and thus a common syntax, we wonder if these resources, developed for slightly different tasks, may be reused for extracting relationships in domain with scarce resources. Several multi-task learning approaches have been proposed to improve performance for a given task using corpus developed for related tasks Collobert et al. (2011). In this paper, we investigate a cross-corpus strategy to improve performances for biomedical RE tasks for which few training data are available, using larger additional corpora

developed for other specific RE tasks. This is done by jointly training deep learning-based models while sharing some of the parameters.

Before or beside deep learning methods, other approaches for RE have been proposed. Co-occurrence-based methods for instance assumes that two entities mentioned frequently in the same unit of text (such as a sentence or a paragraph) are related (Garten & Altman, 2009). Rule-based methods use manually designed, or learned, rules consisting of word morphosyntactic features or sentence-level syntactic features (Fundel et al., 2007). These methods have the advantage of requiring few or no annotated data.

Within machine learning methods, deep learning ones enable to model complex structures such as natural language and successfully applied to various NLP tasks. In particular, it as been successfully applied to RE by training from annotated corpora (Zeng et al., 2014) While other methods mainly depend on the quality of extracted features derived from preexisting NLP systems (*e.g.*, POS tagger, stemmer, lemmatizer or syntactic parser), deep learning models automatically learn lexical features using continuous word vector representations, usually named *word embeddings*, and sentence level features using deep neural network such as Convolutional Neural Network (CNN) (Collobert et al., 2011) or Recursive Neural Networks (RNN) (Socher et al., 2013).These models achieve good performances, but strongly depend on the existence of large training corpora, which make them difficult to use for tasks associated with scarce resources.

In this paper investigate within four specific RE tasks, for which only few training data are available, how large annotated corpora can be used to improve performances of deep neural networks. We experiment two different deep learning approaches that have been previously used for RE. The first is a Multi-Channel CNN (MCCNN)-based model used in (Quan et al., 2016) for biomedical RE and the second is the tree-structured Long Short Term Memory (TreeLSTM) model (Tai et al., 2015), which have been adapted with success for RE (Miwa & Bansal, 2016). The main difference between these two models is the ability of the latter to exploit the syntax of the language by including a dependency tree structure in the vector representation of sentences.

We conduct our experiments using two relatively small biomedical corpora, SNPPhenA and EU-ADR. Both contains less than 400 manually annotated sentences for each task, but note that EU-ADR focus on three different tasks. As supplementary data, we used three larger corpora: SemEval 2013 DDI, ADE and reACE. Details on these five corpora are provided Section 4. Our experiments show that contrary to the MCCNN model, the TreeLSTM model benefit from a cross-corpus learning strategy to improve the RE performances for tasks associated with scarce resources. This is done by training a model with data from two distinct corpora, one small and one large, while sharing the model parameters. In addition, our approach led to state-of-the-art performances for the four biomedical tasks associated with scarce resources.

Section 2 review various deep learning methods used for RE and previous multi-task learning approaches. Section 3 details the MCCNN and TreeLSTM models we use. Section 4 describes corpora used in this study and Section 5 presents our experiments and results. We then conclude with a short discussion section.

2 Related work

2.1 Deep Learning-base Relation extraction

Deep learning models, based on continuous word representations have been proposed to overcome the problem of sparsity inherent to NLP (Huang & Yates, 2009). In Collobert et al. (2011), 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 (Cho et al., 2014), question-answering (Bordes et al., 2014) and RE.

In particular, Zeng et al. (2014) showed that CNN models can also be applied to the task of RE. In this study, they learn a vectorial sentence representation, by applying a CNN model over word and word position embeddings. This representation is then used to feed a softmax classifier (Bishop, 2006). To improve the performance of the RE, other authors consider elements of syntax within the embedding provided to the model: Xu et al. (2015) use the path of grammatical dependencies

between two entities, which is provided by a dependency parsing; Yang et al. (2016) include the relative positions of words in a dependency tree. They also take dependency based context (*i.e.,* child and parent nodes) into account during the convolution.

Beside CNN models that incorporate syntactic knowledge in their embeddings, other approaches go further by proposing neural networks which topology is adapting to the syntactic structure of the sentence. In particular, RNN have been proposed to adapt to tree structures resulting from constituency parsing (Socher et al., 2013; Legrand & Collobert, 2014). In that vein, Tai et al. (2015) introduced a TreeLSTM, 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. (2015). 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 & Bansal (2016) introduced a variant of the TreeLSTM used to compute bidirectional (bottom-up and top-down) tree representations for performing relationship classification. Their model uses different weight matrices depending on whether a node belong to the shortest path or not.

In this paper, we use two deep-learning strategies to address the problem of RE. The first one is a MultiChannel Convolutional Neural Network (MCCNN) introduced in Quan et al. (2016) for biomedical RE. Inspired by the three-channel RGB image processing models, it consider different embedding channels (*i.e.*, different word embeddings versions for each word), allowing to capture different aspects of input words. The second model we used is the TreeLSTM model described in Tai et al. (2015) 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.

2.2 MULTI-TASK LEARNING

Machine learning methods and particularly deep learning ones usually require lots of annotated data in order to obtain reasonable performances. For certain tasks that does not require expert knowledge, such as the recognition of simple objects in an image, gathering lots of annotated data is relatively, easy using for instance crowd-sourcing. Some tasks, such as recognizing a relationship between complex entities that is mentioned in a biomedical scientific publication, are more complex, and the obtention of large corpora in this case can be expensive and time consuming. Several methods have been explored to deal with the lack of training data, such as *bootstrapping* (Jones et al., 1999), which allows accurate training from a small amount of labeled data, along with a large amount of unlabeled data; or *self-training* approaches McClosky et al. (2006) that artificially augment the labeled training set with examples from unlabeled datasets, using labels predicted by the model itself.

Beside, several studies have focused on transferring knowledge acquired from related tasks to help perform a new related task. For instance, Fei-fei et al. (2006) proposed a Bayesian approach to perform *one shot* learning, (*i.e.*, learning to categorize objects from a single example) that takes advantage of knowledge coming from previously learned categories.

Multi-task Learning is a learning approach in which performances on a given task are improved using information contained in the training signals of auxiliary related tasks Caruana (1997). It is a form of inductive transfer where the auxiliary task introduce an inductive bias during training. This is usually done by training tasks in parallel while using a shared representation (Sutton et al., 2007; Ando & Zhang, 2005). In Collobert et al. (2011), the authors jointly trained a CNN on various natural language processing tasks including part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. They showed that sharing a portion of the network weights during training led to better performances for all the individual tasks.

3 MODELS

We consider in this article MCCNN and TreeLSTM models that both compute a fixed-size vector representation for the whole sentence by composing input embeddings. A score is then computed 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).

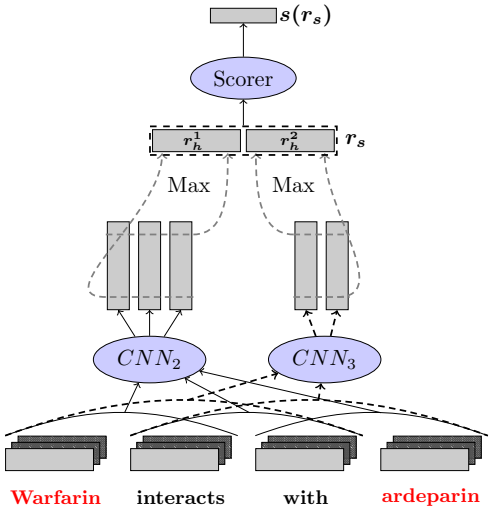

Figure 1: Visualization of the MCCNN model with three channels and two CNN with kernels of size 2 and 3 respectively. CNN_i denotes a CNN with a kernel size i. Red words correspond to the entities.

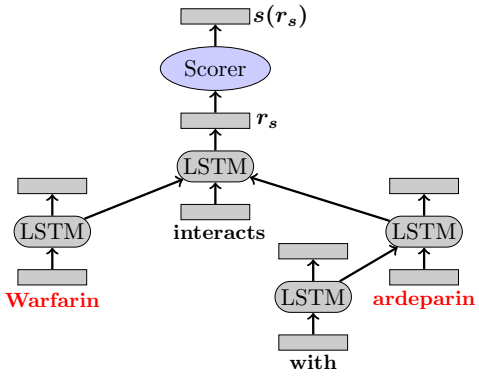

Figure 2: Visualization of the TreeLSTM model. Red words correspond to the entities.

In this section, we first introduce the embedding input layer, which in common to both approaches (*i.e.*, MCCNN and TreeLSTM); Then, we detail how each approach composes sequences of embedding in order to compute an unique vectorial sentence representation; Finally, we present the scoring layer, which is common to both approaches.

3.1 Input Layer

Both models are fed with *word embeddings* (*i.e.*, continuous vectors) of dimension d_w, along with extra *entity embeddings* of size d_e, which are concatenated to word embeddings. Formally, given a sentence of N words, $w_1, w_2, \ldots, w_N$, each word $w_i \in W$ is first embedded in a d_w-dimensional vector space by applying a lookup-table operation:

$$LT_W(w_i) = W_{w_i} \ ,$$

where the matrix $W \in R^{d_w \times |W|}$ represents the parameters to be trained in this lookup layer. Each column $W_{w_i} \in R^{d_w}$ corresponds to the vector embedding of the $w_i{}^{th}$ word in our dictionary W.

Three entity embeddings (coming from a simple 3-elements dictionary) enable to distinguish between words which compose either the first entity, the second entity or are not part of any entity in a sentence. They are respectively called *first entity*, *second entity* and *other* embeddings. Finally, word and entity embeddings are concatenated to form the input corresponding to a given word. Let's denote x_i the concatenated input corresponding to the ith word.

3.2 Composition Layers

Both models take the embeddings as input and output a fixed-size representation r_s of size d_s. This section details the two models used in this study.

3.2.1 MCCNN

The MCCNN models applies a variable kernel size CNN to multiple input channels of word embeddings. More formally, given an input sequence $x_1, \ldots, x_N$, applying a kernel of size k to the ith window is done using the following formula:

$$C = h(\sum_{j=1}^{c} W[x_{\frac{i-1}{2}}, \ldots, x_i, \ldots, x_{\frac{i+1}{2}}]^j + b)$$

where $[\,]^j$ denotes the concatenation of inputs from channel j, $W \in \mathcal{R}^{(d_w+d_e)\times d_h}$ and $b \in \mathcal{R}^{d_h}$ are the parameters , h is a pointwise non-linear function such as the hyperbolic tangent and c is the number of input channels. Inputs with indices exceeding the input boundaries ($\frac{i-1}{2} < 1$ or $\frac{i+1}{2} > N$) are mapped to a special padding vector (which is also learned). A fixed size representation $r_h \in \mathcal{R}^{d_h}$ is then obtain by applying a max-pooling over time:

$$r_h = \max C$$

We denote K the number of kernel with different sizes. A sentence representation $r_s \in \mathcal{R}^{d_s}$ (with $d_s = K * d_h$) is finally obtained by concatenating the output corresponding to the K kernels

$$r_s = [r_h^1, \ldots, r_h^k] \ ,$$

where r_h^k correspond to the output of the kth kernel. Figure 1 illustrates the structure of a two-channel CNN, with two kernels of size 2 and 3, on a four-words sentence.

3.2.2 TREELSTM

The TreeLSTM model (Tai et al., 2015) 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:

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

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

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 kth child. Each TreeLSTM 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. The TreeLSTM outputs a sentence representation $r_s \in \mathcal{R}^{d_s}$ corresponding to the output state o_j of the top tree node (*i.e.*, the *root* node of the dependency tree that spans all the others). Figure 2 illustrates the structure of the TreeLSTM computed for a four-words sentence.

3.3 SCORING LAYER

Both the MCCNN and the TreeLSTM models output an unique vector representation $r_s \in \mathcal{R}^{d_s}$ that takes the entire sentence into account. This representation is used to feed a single layer neural network classifier, which outputs a score vector with one score for each possible type of relationships. This vector of scores is obtained using the following formula:

$$s(r_s) = W^{(s)} r_s + b^{(s)} \ ,$$

where $W^{(s)} \in \mathcal{R}^{d_s \times |S|}$ and $b^{(s)} \in \mathcal{R}^{|S|}$ are the trained parameters of the scorer, $|S|$ is the number of possible relations. The scores are interpreted as probabilities using a softmax layer (Bishop, 2006).

4 DATASETS

We explore how RE tasks that focus on a type of relationship associated with scarce resources may take advantage from existing corpora, in other words how completing a small training corpus with a larger one may help the RE task when the latter is annotated with a different type of relationships. For this purpose, we selected *(i)* two small biomedical corpora, SNPPhenA and the EU-ADR corpus and *(ii)* three larger corpora, the SemEval 2013 DDI corpus, the ADE corpus and the reACE corpus. These corpora are publicly available and detailed in the following section. Table 4.2 summarizes the main characteristics of these five corpora and the following section details them.

4.1 SMALL CORPORA

- **SNPPhenA** (Bokharaeian et al., 2017) is a corpus of abstracts of biomedical publications, obtained from PubMed[1], annotated with two types of entities: *single nucleotide polymorphisms* (SNPs) and *phenotypes*. Relationships between these entities are annotated and classified in 3 categories: *positive*, *negative* and *neutral* relationships. The *neutral* relationship type is used when no relationship is mentioned in the sentence between two annotated entities.
- **EU-ADR** (Van Mulligen et al., 2012) is a corpus of abstracts obtained from PubMed and annotated with *drugs*, *disorders* and drug targets (*proteins/genes* or *gene variants*) entities. It is composed of 3 subcorpora, focusing either on target-disease, target-drug or drug-disease relationships. Each of them consist of 100 abstracts. Annotated relationships are classified in 3 categories: *positive*, *speculative* and *negative associations* (PA, SA and NA respectively). In Bravo et al. (2015), performances are assessed over the `TRUE` class, which is composed of the classes PA, SA and NA, in contrast with the `FALSE` class composed of sentences where two entities are co-occurring, but without relationship annotated between them.

4.2 LARGE CORPORA

- **SemEval 2013 DDI** (Herrero-Zazo et al., 2013) 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) (Gurulingappa et al., 2012) consists of MEDLINE case reports, annotated with *drug* and *conditions* (*e.g.*, diseases, signs and symptoms) along with untyped relationships between them, when one is mentioned.
- **reACE** (Edinburgh Regularized Automatic Content Extraction) (Hachey et al., 2012) consists of English broadcast news and newswire annotated with *organization*, *person*, *fvw* (facility, vehicle or weapon) and *gpl* (geographical, political or location) entities along with relationships between them. Relationships are classified in five categories (*general-affiliation*, *organisation-affiliation*, *part-whole*, *personal-social* and *agent-artifact*).

5 EXPERIMENTS

5.1 TRAINING AND EXPERIMENTAL SETTINGS

Following Collobert et al. (2011), our models are trained by minimizing a log-likelihood function over the training data. All parameters, including weights, biases and embeddings were updated via Backpropagation for the MCCNN and Backpropagation through Structure (BPTS) (Goller & Kuchler, 1996) for the TreeLSTM.

All the hyper-parameters were tuned using a 10 fold cross-validation. Word embeddings were pre-trained on PubMed abstracts using the method described in Lebret & Collobert (2013)

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

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
reACE	–	5,984	2,486	–	–	4	5

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, ADR-EXT and reACE have no proper test corpus.

MCCNN model. Following Kim (2014) both channels are initialized with pre-trained word embeddings but gradients were back-propagated only through one of the channels. Hyper-parameters were fixed to $d_w = 100$, $d_e = 10$, $d_h = 100$ and $d_s = 200$. We applied a dropout regularization after the embedding layers.

TreeLSTM model. Dependency trees were obtained using the Stanford Parser (Chen & Manning, 2014). Hyper-parameters were fixed to $d_w = 100$, $d_e = 10$, $d_h = 200$ and $d_s = 200$. We applied a dropout regularization (Srivastava et al., 2014) after every TreeLSTM unit and after the embedding layers. The drop probability for each connexion was fixed to 0.25. All the parameters are initialized randomly except the word embeddings.

We evaluated performances in terms of precision (P), recall (R) and f-measure (F). For multi-label classifications, 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. For each of the 10 folds, we performed 10 experiments starting from different random weight initializations. Thus, each result is an average of 100 experiments.

5.2 Cross-corpus study

In this subsection, we present our cross-corpus training strategy and its results. For each cross-corpus experiments, the same network was used for the different corpora (*i.e.*, same embedding layer and TreeLSTM weights), except for the scorer, which is different for each corpus as the number and types of relationships 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 for the TreeLSTM model, additional data consistently improved the performances. More interestingly, this phenomenon occurred even for corpora with different types of entities such as the combination of SNPPhenA and SemEval 2013 DDI and, to a lesser extend, for a corpus outside of the biomedical domain (reACE). This phenomenon was not observed for the MCCNN model for which performance tended to decrease slightly when using the cross-corpus learning strategy.

5.3 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 tasks associated respectively with SNPPhenA and EU-ADR, respectively reported in Bokharaeian et al. (2017) and Bravo et al. (2015). Our results are obtained using, for each fold, an ensemble of the 5 best models (according to the validation) starting from different random initialization, following Legrand & Collobert (2014). We report the 10 folds average performance. 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 four EU-ADR subtasks.

Test Corpus (Model)	Train corpus	P	R	F	σ_F
SNPPhenA (TreeLSTM)	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
	+ reACE	61.8	74.3	67.1	0.036
SNPPhenA (MCCNN)	SNPPhenA	55.1	75.0	63.3	0.048
	+ SemEval 2013 DDI	55.3	74.4	63.3	0.049
	+ ADE-EXT	56.1	73.2	63.2	0.048
	+ reACE	53.2	70.9	60.6	0.041
EU-ADR drug-disease (TreeLSTM)	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
	+ reACE	74.3	91.1	79.3	0.143
EU-ADR drug-disease (MCCNN)	EU-ADR drug-disease	73.3	94.7	80.2	0.142
	+ SemEval 2013 DDI	72.6	87.9	76.6	0.143
	+ ADE-EXT	73.0	85.5	76.0	0.145
	+ reACE	74.1	91.5	79.2	0.138
EU-ADR drug-target (TreeLSTM)	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
	+ reACE	70.4	96.5	80.8	0.093
EU-ADR drug-target (MCCNN)	EU-ADR drug-target	74.5	92.3	81.0	0.093
	+ SemEval 2013 DDI	74.9	88.8	80.0	0.106
	+ ADE-EXT	76.3	87.4	80.3	0.101
	+ reACE	73.4	92.1	80.5	0.078
EU-ADR target-disease (TreeLSTM)	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 (TreeLSTM)	77.7	89.5	83.3	0.069
	+ reACE	75.89	91.7	83.0	0.077
EU-ADR target-disease (MCCNN)	EU-ADR target-disease ( )	76.9	91.8	82.6	0.077
	+ SemEval 2013 DDI	77.6	90.6	82.5	0.071
	+ ADE-EXT	75.5	87.4	81.8	0.101
	+ reACE	77.1	91.2	82.0	0.068

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

Test corpus	Train corpus	P	R	F
SNPPhenA	**Bokharaeian et al. (2017)**			
	SNPPhenA	56.6	59.8	58.2
	This work			
	SNPPhenA + ADE-EXT	64.5	75.2	**69.4**
EU-ADR drug-disease	**Bravo et al. (2015)**			
	EU-ADR drug-disease	70.2	93.2	79.3
	This work			
	EU-ADR drug-disease + SemEval 2013 DDI	74.8	90.6	**82.0**
EU-ADR drug-target	**Bravo et al. (2015)**			
	EU-ADR drug-target	74.2	97.4	83.0
	This work			
	EU-ADR drug-target + SemEval 2013 DDI	73.5	95.6	**83.1**
EU-ADR target-disease	**Bravo et al. (2015)**			
	EU-ADR target-disease	75.1	97.7	84.6
	This work			
	EU-ADR target-disease + SemEval 2013 DDI	78.7	91.4	**84.6**

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

6 DISCUSSION

Results presented in Table 2 show that, in our settings, the TreeLSTM model benefits from a cross-corpus learning strategy, while it is useless, or sometimes counterproductive for the MCCNN model. One may think that the TreeLSTM model, due to its ability to exploit the syntactic structure of the sentence, is better at understanding the sentences from the small datasets by exploiting the syntactic patterns observed in the additional data. This idea is reinforced by the fact that even a corpus that does not share the same entities nor a close vocabulary, such as reACE in which no biomedical vocabulary appear, can be helpful for biomedical RE. This assessment could be interestingly explored in further work.

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 suitable 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 types of texts in sources (*e.g.*, medical case report for ADE-EXT, news for reACE, research articles for the others). Higher level syntactic analysis (such as the average distance between the two entities or the nature of the lowest common ancestor in the dependency graph) could provide insights on this question, and help in characterizing the right corpus to select for a cross-corpus training.

7 CONCLUSION

In this paper, we empirically demonstrated that a cross-corpus learning strategy can be beneficial to tackle biomedical RE tasks for which few annotated resources are available, when using the TreeLSTM model. Interestingly, we showed that any additional corpus, even when focusing on unrelated domain can carry useful information and lead to improved performances. Additionally, the cross-corpus approach led to state-of-the-art performances for 4 biomedical RE task focusing on SNP-phenotype, target-disease, target-drug and drug-disease relationships. We think that cross-corpus training could be reproduced and thus valuable in other specialized domains in which training resources are scarce.

REFERENCES

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 2005.

Christopher M Bishop. *Pattern recognition and machine learning*. 2006.

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.

Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.

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.

Rich Caruana. Multitask learning. *Mach. Learn.*, 1997.

Danqi Chen and Christopher D. Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, 2014.

Kyunghyun Cho, Bart Van Merriënboer, et al. On the properties of neural machine translation: Encoder-decoder approaches. 2014.

Ronan Collobert, Jason Weston, et al. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2011.

Li Fei-fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 28:2006, 2006.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 2007.

Yael Garten and Russ B. Altman. Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text. *BMC Bioinfo.*, 2009.

Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of ICNN*, 1996.

Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. Extraction of potential adverse drug events from medical case reports. *Journal of Biomedical Semantics*, 2012.

B. Hachey, C. Grover, and R. Tobin. Datasets for generic relation extraction. *Nat. Lang. Eng.*, 2012.

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.

Fei Huang and Alexander Yates. Distributional representations for handling sparsity in supervised sequence-labeling. In *Proceedings of ACL*, 2009.

Rosie Jones, Andrew McCallum, Kamal Nigam, and Ellen Riloff. Bootstrapping for text learning tasks. In *IN IJCAI-99 WORKSHOP ON TEXT MINING: FOUNDATIONS, TECHNIQUES AND APPLICATIONS*, 1999.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014.

Rémi Lebret and Ronan Collobert. Word emdeddings through hellinger pca. *EACL*, 2013.

Joël Legrand and Ronan Collobert. Joint RNN-based greedy parsing and word composition. *arXiv preprint arXiv:1412.7028*, 2014.

Yang Liu, Furu Wei, et al. A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646*, 2015.

David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

Makoto Miwa and Mohit Bansal. End-to-end relation extraction using LSTMs on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

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

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. Parsing with compositional vector grammars. In *Proceedings of ACL*, 2013.

Nitish Srivastava, Geoffrey E Hinton, et al. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 2007.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured LSTM networks. In *ACL*, 2015.

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.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *EMNLP*, 2015.

Yunlun Yang, Yunhai Tong, et al. A position encoding convolutional neural network based on dependency tree for relation classification. In *EMNLP*, 2016.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. In *proceedings of COLING*, 2014.