Syntax-based Transfer Learning for the Task of Biomedical Relation Extraction

Joël Legrand, Yannick Toussaint, Chedy Raïssi, Adrien Coulet

Syntax-based Transfer Learning for the Task of Biomedical Relation Extraction

Joël Legrand[1], Yannick Toussaint[1], Chedy Raïssi[1] and Adrien Coulet[1,2]

[1] Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
[2] Stanford University, Stanford Center for Biomedical Informatics Research, California
joel.legrand@loria.fr

Abstract

Transfer learning (TL) proposes to enhance machine learning performance on a problem, by reusing labeled data originally designed for a related problem. In particular, domain adaptation consists, for a specific task, in reusing training data developed for the same task but a distinct domain. This is particularly relevant to the applications of deep learning in Natural Language Processing, because those usually require large annotated corpora that may not exist for the targeted domain, but exist for side domains. In this paper, we experiment with TL for the task of Relation Extraction (RE) from biomedical texts, using the TreeLSTM model. We empirically show the impact of TreeLSTM alone and with domain adaptation by obtaining better performances than the state of the art on two biomedical RE tasks and equal performances for two others, for which few annotated data are available. Furthermore, we propose an analysis of the role that syntactic features may play in TL for RE.

1 Introduction

A bottleneck problem for training deep learning-based architecture on text is the availability of large enough annotated training corpora. This is especially an issue in highly specialized domains such as those of biomedicine. TL approaches address this problem by leveraging existing labeled data originally designed for related tasks or domains (Weiss et al., 2016). However, adaptation between dissimilar domains may lead to negative transfer, *i.e.* transfer that decreases the performance for the target domain. In this article, we apply a TL strategy using the TreeLSTM model for the task of biomedical Relation Extraction (RE). We propose an analysis of the syntactic features of source and target domain corpora to provide elements of interpretation for the improvements we obtained.

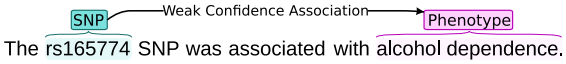


Figure 1: Example of relationship typed as *Weak Confidence Association* between two named entities: a *SNP (single nucleotide polymorphism)* and a *Phenotype*, from the SNPPhenA corpus.

Relation Extraction (RE) aims at identifying in raw and unstructured text all the instances of a predefined set of relations between identified entities. A relationship takes the form of an edge between two or more named entities as illustrated in Figure 1. We are considering here binary RE that can be seen as a classification task by computing a score for each possible relation type, given a sentence and two identified entities.

Deep learning methods have demonstrated good ability for RE (Zeng et al., 2014), but one of their drawbacks is that, in order to obtain reasonable performances, they generally require a large amount of training data, *i.e.*, text corpora where entities and relationships between them are annotated. The assembly of this kind of domain- and task-specific corpora, such as those of interest in biomedicine, is time consuming and expensive because it involves complex entities (*e.g.*, genomic variations, complex phenotypes), complex relationships (which may be hypothetical, contextualized, negated, *n*-ary) and requires trained annotators. This explains why only few and relatively small (*i.e.*, few hundreds of sentences) corpora are available for some biomedical RE tasks, making these resources particularly valuable. Distinct approaches, such as TL or *distant supervision* (Mintz et al., 2009) have been particularly explored to overcome this limit. With the latter approach, existing relationships available in knowledge- or data-bases are used to enrich the training set, with-

out considering more labeled corpora .

Domain adaptation is a type of TL that allows taking advantage of data annotated for a *source* domain to improve the performances in a related *target* domain (Weiss et al., 2016). However, even if the source and target domain share the same language (*i.e.*, English), thus a common syntax, TL between domains may lead to negative transfer since specific source domains may use specific vocabularies as well as specific formulations that are inadequate to the target domain. Hence, we need to better understand and characterize what makes a source corpus potentially helpful, or harmful, with regard to a RE task.

The contribution of this paper is twofold. First, we show that, compared to a baseline Convolutional Neural Network (CNN)-based model, a syntax-based model (*i.e.*, the TreeLSTM model) can better benefit from a TL strategy, even with very dissimilar additional source data. We conduct our experiments with two biomedical RE tasks and relatively small associated corpora, SNPPhenA (Bokharaeian et al., 2017) and EU-ADR (van Mulligen et al., 2012) as target corpora and three larger RE corpora, Semeval 2013 DDI (Herrero-Zazo et al., 2013), ADE-EXT (Gurulingappa et al., 2012), reACE (Hachey et al., 2012) as source corpora. Second, we propose a syntax-based analysis, using both quantitative criteria and qualitative observations, to better understand the role of syntactic features in the TL behavior.

2 Related work

2.1 Deep Learning Models for Relation Extraction

Deep learning models, based on continuous word representations have been proposed to overcome the problem of sparsity inherent to NLP (Huang and Yates, 2009). In Collobert et al. (2011), the authors proposed a unified CNN architecture to tackle various NLP problems traditionally handled with statistical approaches. They obtained state-of-the-art performances for several tasks, while avoiding the hand design of task specific features.

Zeng et al. (2014) showed that CNN models can also be applied to RE. In this study, they learn a vectorial sentence representation, by applying a CNN model over word and word position embeddings, which is used to feed a softmax classifier (Bishop, 2007). To improve the performance of RE, authors, such as Xu et al. (2015) and Yang et al. (2016), consider elements of syntax within the embedding provided to the model.

Beside CNN models that incorporate syntactic knowledge in their embeddings, other approaches proposed neural networks (NN) in which the topology is adapted to the syntactic structure of the sentence. In particular, Recursive Neural Network (RNN) have been proposed to adapt to tree structures resulting from constituency parsing (Socher et al., 2013; Legrand and Collobert, 2014). In this vein, Tai et al. (2015) introduced a TreeLSTM, a generalization of LSTM for tree-structured network topologies, which allows processing trees with arbitrary branching factors.

The first model to use RNN for RE was proposed by Liu et al. (2015). The authors introduced a CNN-based model applied on the shortest dependency path between two entities, augmented with a RNN-based feature designed to model subtrees attached to the shortest path. Miwa and Bansal (2016) introduced a variant of the TreeLSTM that allows, like the model used in this paper, to take the whole dependency tree into account and not only the shortest path between two entities.

In this paper, we compare two deep learning strategies for RE: (1) the MultiChannel CNN (MCCNN) model (Quan et al., 2016), which has been successfully applied to the task of protein-protein interaction extraction without using any syntactic feature as input and (2) the TreeLSTM model (Tai et al., 2015), which is designed for considering dependency trees. These two models are detailed in section 3.

2.2 Transfer learning

TL allows to overcome the lack of training data for a given *target* task by transferring knowledge from *source* data not originally designed for that purpose (Weiss et al., 2016). One can distinguish *multitask learning* in which performances on a given task are improved using information contained in the training signals of auxiliary related tasks (Caruana, 1997) from *domain adaptation* in which only one task is considered but its application domains differ (Ben-David et al., 2010). While the former is a form of inductive transfer in which the auxiliary task introduces an inductive bias during training, the latter is a form of transductive transfer.

Domain adaptation approaches have been proposed for RE, including kernel based methods

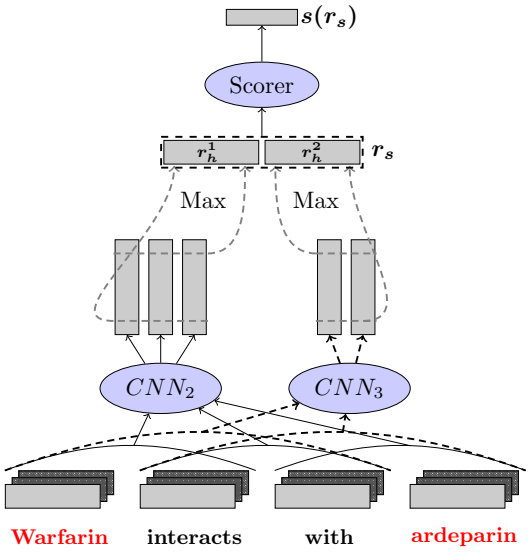


Figure 2: The MCCNN model with three channels, two CNN kernels of size 2 (CNN_2) and 3 (CNN_3). Red words correspond to the entities.

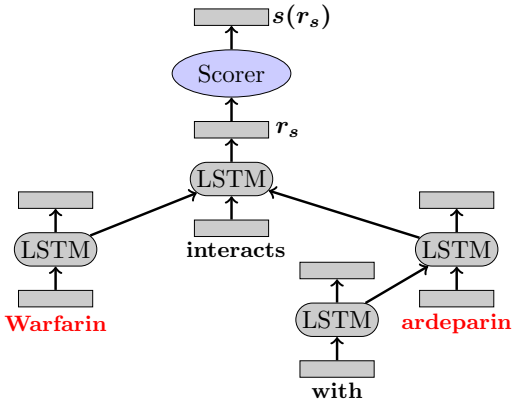


Figure 3: The TreeLSTM model. Each node takes as input the representation of its children. Red words correspond to the entities.

such as Plank and Moschitti (2013) focusing on unsupervised domain adaptation (*i.e.*, without any labeled target data) and deep learning based ones such as (Fu et al., 2017; Zhao et al., 2017) focusing on domain adversarial learning (an approach which ensures that the feature distributions over the source and target domains are made similar using an extra domain classifier at train time). Differently, our approach is a case of multi-source domain adaptation (*i.e.*, implying that we have labeled data, both in target and source corpora) and does not involve adversarial training.

Negative transfer occurs when the information learned from a source domain and task has a negative impact on the performances of the target task. Despite the fact that negative transfer is a major issue in TL, to our knowledge only few works have been conducted to overcome this problem (Weiss et al., 2016). Most of them use a relatedness metrics to select the elements of the source that are the most related to the target. For instance, Seah et al. (2013) defined a positive transferability measure that allows removing irrelevant source data. Ge et al. (2014) also focused on domain adaptation from multiple sources. They proposed a method to avoid negative learning caused by unrelated or irrelevant source domains, using a weighting mechanism based on a relatedness metrics between the source and target data.

In this work, we experiment with a domain adaptation method on the RE task using the TreeLSTM model, with relatively small biomedical corpora as target corpora and, larger biomedical or general domain corpora as source corpora. We also provide elements of interpretation of the impact of syntactic dependency structures on TL. In this matter, and unlike Seah et al. (2013) or Ge et al. (2014), the relatedness measures used in this work emphasizes the key role of syntax in TL with TreeLSTM.

3 Models

We compare in this article the performances of the MCCNN and TreeLSTM models. Both models compute a fixed-size vector representation for a whole sentence by composing input embeddings. A score is computed for each possible type of relationship (*e.g.*, negative, positive or speculative) between two identified entities.

In this section, we first introduce the embedding input layer, which is common to both approaches (*i.e.*, MCCNN and TreeLSTM); Then, we detail how each approach composes sequences of embedding in order to compute a 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. These embeddings are concatenated to form the input of the model. Formally, given a sentence of N words, $w_1, w_2, \ldots, w_N$, each word $w_i \in \mathcal{W}$ is first embedded in a d_w-dimensional vector space by ap-

	Corpus name	Subcorpus	Train Size sent.	Train Size rel.	Test Size sent.	Test Size rel.	#Entity Types	#Relation Types
Target corpora	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
Source corpora	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 our target and source corpora. Two corpora are divided into subcorpora. The sizes of the training and test corpora are reported in term of number of sentences (sent.) and annotated relationships (rel.). EU-ADR, ADR-EXT and reACE have no proper test corpus.

plying a lookup-table operation: $LT_W(w_i) = W_{w_i}$, where the matrix $W \in R^{d_w \times |\mathcal{W}|}$ represents the parameters to be trained in this lookup-table layer. The dictionary $\mathcal{W}$ is composed of all the words of the given corpus. Each column $W_{w_i} \in R^{d_w}$ corresponds to the vector embedding of the $w_i{}^{th}$ word in our dictionary $\mathcal{W}$.

Besides, 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. 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 i^{th} word.

3.2 Composition layers

Both models take the embeddings as input and output a fixed-size representation r_s of size d_s, which corresponds to the whole sentence with two identified entities. Accordingly, one sentence with more than two entities will lead to one embedding for each pair of entities. 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. Inspired by the three-channel RGB image processing models, it considers different embedding channels (i.e., different word embeddings versions for each word) allowing to capture different aspects of input words.

More formally, given an input sequence $x_1, \ldots, x_N$, applying a kernel to the i^{th} window of size k is done using the following formula:

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

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, d_h is the size of the hidden layer, h is a pointwise non-linear function such as the hyperbolic tangent and c is the number of input channels. For each kernel, a fixed size representation $r_h \in \mathcal{R}^{d_h}$ is then obtained by applying a max-pooling over time (here, the time means the position in the sentence).: $r_h = \max C$

We denote K the number of kernels 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 corresponds to the output of the k^{th} kernel. Figure 2 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, and more specifically its *Child-Sum* version, (Tai et al., 2015) processes the dependency tree associated with an input sentence in a bottom-up manner. This model is suitable for processing dependency trees since it handles trees with arbitrary branching factors and no order between children of a node. This is done by recursively processing the nodes of the tree, using at each iteration, the representations of the children of the current node as input. The transition function for a node j and a set of children $C(j)$ can be found in the original paper (Tai et al., 2015) using $x_j \in \mathcal{R}^{d_w+d_e}$ as input for node j. 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 3 illustrates the structure of the TreeLSTM computed for a four-words sentence.

3.3 Scoring layer

Both the MCCNN and TreeLSTM models output a unique vector representation $r_s \in \mathcal{R}^{d_s}$ that takes the entire sentence into account, as well as two identified entities. This representation is used to feed a single layer NN classifier, which outputs a score vector with one score for each possible type of relationship. This vector is obtained using the 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 relation types. The scores are interpreted as probabilities using a softmax layer (Bishop, 2007).

4 Datasets

We explore how RE tasks that focus on a type of relationship associated with scarce resources may take advantage from larger corpora developed for distinct domains. To this purpose, we selected *(i)* two small *target* biomedical corpora and *(ii)* three larger *source* corpora. All are publicly available and detailed in the following section. Table 3 summarizes their main characteristics.

4.1 Target corpora

SNPPhenA (Bokharaeian et al., 2017) is a corpus of abstracts of biomedical publications, obtained from PubMed (Fiorini et al., 2017), annotated with two types of entities: *single nucleotide polymorphisms* (SNPs) and *phenotypes*. Relationships between them are annotated and classified in 3 types: *positive*, *negative* and *neutral*.

EU-ADR (van Mulligen et al., 2012) is a corpus of PubMed abstracts annotated with *drugs*, *diseases* and drug targets (*proteins/genes* or *gene variants*) entities. It is composed of 3 subcorpora of 100 abstracts each, encompassing annotations of either target-disease, target-drug or drug-disease relationships. Annotated relationships are classified in 3 types: *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 PA, SA and NA types, in contrast with the `FALSE` class.

4.2 Source corpora

SemEval 2013 DDI (Drug-Drug Interaction) (Herrero-Zazo et al., 2013) consists of texts from DrugBank and MEDLINE annotated with drugs. Drug are categorized in 4 categories: *drug*, *brand*, *group* and *drug_n* (*i.e.*, active substances not approved for human use). Relationships are classified in 4 types: *mechanism*, *effect*, *advice* and *int* (default category, when no detail is provided).

ADE-EXT (Adverse Drug Effect corpus, extended) (Gurulingappa et al., 2012) consists of MEDLINE case reports, annotated with *drugs* and *conditions* (*e.g.*, diseases, signs and symptoms), along with untyped relationships between them.

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 types: *general-affiliation*, *organisation-affiliation*, *part-whole*, *personal-social* and *agent-artifact*.

5 Experiments

5.1 Training and Experimental Settings

Our models were trained by minimizing the log-likelihood over the training data. All parameters (weights, biases and embeddings) were iteratively updated via backpropagation for the MCCNN and backpropagation Through Structure (Goller and Kuchler, 1996) for the TreeLSTM. Hyper-parameters were tuned using a 10-fold cross-validation by selecting the values leading to the best averaged performance, and fixed for the remaining experiments. Word embeddings were pre-trained on ~3.4 million PubMed abstracts (corresponding to all those published between Jan. 1, 2014 and Dec. 31, 2016) using the method described in Lebret and Collobert (2014).

MCCNN model. Following Kim (2014) both channels were initialized with pre-trained word embeddings, but gradients were backpropagated only through one of the channels. Hyper-parameters were fixed to $d_w = 100$, $d_e = 10$, $d_h = 100$ for each of the 2 channels, $d_s = 2 \times d_h = 200$. We used two kernels of size 3 and 5 respectively. We applied a dropout regularization after the embedding layers (Srivastava et al., 2014) with a dropout probability fixed to 0.25.

Test Corpus	Model	Train corpus	P	R	F	σ_F
SNPPhenA	TreeLSTM	SNPPhenA alone	58.9	73.8	65.5	4.1
		+ SemEval 2013 DDI	65.2	71.1	**68.0**	4.7
		+ ADE-EXT	62.8	72.1	67.2	3.4
		+ reACE	61.8	74.3	67.1	3.6
	MCCNN	SNPPhenA alone	55.1	75.0	**63.3**	4.8
		+ SemEval 2013 DDI	55.3	74.4	**63.3**	4.9
		+ ADE-EXT	56.1	73.2	63.2	4.8
		+ reACE	53.2	70.9	60.6	4.1
EU-ADR drug-disease	TreeLSTM	EU-ADR drug-disease alone	74.8	84.1	79.1	12.3
		+ SemEval 2013 DDI	74.8	90.6	**82.0**	13.1
		+ ADE-EXT	73.9	88.2	80.4	13.7
		+ reACE	74.3	91.1	79.3	14.3
	MCCNN	EU-ADR drug-disease alone	73.3	94.7	**80.2**	14.2
		+ SemEval 2013 DDI	72.6	87.9	76.6	14.3
		+ ADE-EXT	73.0	85.5	76.0	14.5
		+ reACE	74.1	91.5	79.2	13.8
EU-ADR drug-target	TreeLSTM	EU-ADR drug-target alone	72.4	90.6	80.2	10.9
		+ SemEval 2013 DDI	71.9	95.5	**82.5**	8.5
		+ ADE-EXT	70.2	96.7	80.9	9.2
		+ reACE	70.4	96.5	80.8	9.3
	MCCNN	EU-ADR drug-target alone	74.5	92.3	**81.0**	9.3
		+ SemEval 2013 DDI	74.9	88.8	80.0	10.6
		+ ADE-EXT	76.3	87.4	80.3	10.1
		+ reACE	73.4	92.1	80.5	7.8
EU-ADR target-disease	TreeLSTM	EU-ADR target-disease alone	77.0	89.7	82.7	6.4
		+ SemEval 2013 DDI	77.4	91.6	**83.9**	8.2
		+ ADE-EXT	77.7	89.5	83.3	6.9
		+ reACE	75.9	91.7	83.0	7.7
	MCCNN	EU-ADR target-disease alone	76.9	91.8	**82.6**	7.7
		+ SemEval 2013 DDI	77.6	90.6	82.5	7.1
		+ ADE-EXT	75.5	87.4	81.8	10.1
		+ reACE	77.1	91.2	82.0	6.8

Table 2: Results of our TL strategy in terms of precision (P), recall (R) and f-measure (F). σ_F is the standard deviation of the f-measure. The + in the column *Train corpus* indicates that we trained our model using the target corpus plus one additional source corpus.

TreeLSTM model. Dependency trees were derived from parsing trees obtained using the Charniak-Johnson parser trained on GENIA and PubMed data (McClosky and Charniak, 2008). Hyper-parameters were fixed to $d_w = 100$, $d_e = 10$, $d_h = 200$ and $d_s = 200$. We applied a dropout regularization after every TreeLSTM unit and after the embedding layers. The dropout probability 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[1]. For SNPPhenA, we performed a cross-validation using 10% of the corpus for the validation and the provided test corpus for testing (which is about 30% the size of the training corpus). Because no 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.

[1]The macro-average metric is less impacted by classes with few test instances (and thus a high variance). For this reason, it is more representative of the performance of our model.

5.2 Transfer learning experiment

In this subsection, we present our TL strategy and its results. Following a standard practice in deep learning, the transfer learning is done by training models in parallel while using shared representations, as illustrated by (Collobert et al., 2011). In other terms, for each experiment, the same network, initialized with random weights, is used for each corpus (i.e., same embedding layer and TreeLSTM weights), except for the scorer, which is adapted to each corpus as the number and types of relationships may change. During the training phase, using a standard stochastic gradient descent procedure (Robbins and Monro, 1985), we randomly pick training sentences from the mixed corpus (i.e., target + one source training corpora).

Test corpus	Work (train corpus)	P	R	F
SNPPhena	**Bokharaeian et al. (2017)** (SNPPhenA)	56.6	59.8	58.2
	This work (SNPPhenA + SemEval 2013 DDI)	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	**83.3**
	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 between the state of the art (Bokharaeian et al., 2017; Bravo et al., 2015) and this work in terms of precision (P), recall (R) and F-measure (F). Results reported for this work are ensembles of the 5 best models obtained.

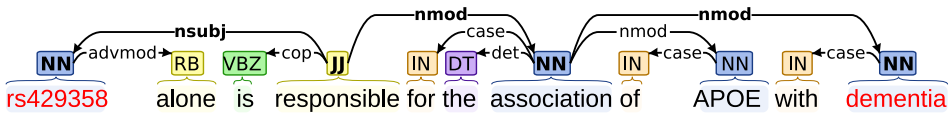


Figure 4: Dependency parse tree of a sentence from SNPPhena expressing a relation between the entities *rs429358* and *dementia*. The shortest dependency path between the two entities is shown in bold.

This training procedure is done, starting from different random initialization for each fold of our cross-validation. Table 2 presents the results of the TL study. Each results is an average of 100 experiment (10 experiments for each fold starting from different random initialization). We observed that for the TreeLSTM model, additional source corpora consistently improved the performances. More interestingly, this phenomenon occurs even for corpora of distinct types of entities such as the combination of SNPPhenA and SemEval 2013 DDI and, to a lesser extend, with the corpus that is outside of the biomedical domain, reACE. We note that the pre-trained embeddings were obtained using biomedical sources. This may affect the TL performance with reACE that is not of the biomedical domain. Also, we did not observed any benefit of the TL strategy for the MCCNN model, which performances decrease slightly in comparison with the baseline experiments.

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 (Bokharaeian et al., 2017) and EU-ADR (Bravo et al., 2015). Our results are obtained using, for each fold, an ensemble of the 5 best models for this fold, according to the validation. The ensembling was done by averaging the scores $s(r_s)$ of each individual model, following Legrand and 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 one EU-ADR subtasks and lead to similar performances for the two remaining EU-ADR subtasks.

6 On the role of syntactic features in transfer learning

Empirical results suggest that the TreeLSTM model is more positively-influenced by syntactic similarity between source and target corpora than by domain closeness. Indeed, the TreeLSTM model explicitly includes the syntactic structure of the sentences in the network topology. Thus, a source corpus, such as reACE, that share neither entity nor vocabulary with the target corpus proved to be helpful. We propose in the following an analysis of the role of the syntactic features. We also provide real examples illustrating similarities between corpora and comment them.

Syntactic features. We propose three comparisons based on patterns extracted from shortest paths between two entities in dependency graphs. Shortest path proved to be effective for RE (Bunescu and Mooney, 2005; Cellier et al., 2010). From a shortest path (as between *rs429358* and *dementia* in Figure 4), we extract 3 different patterns. The first one is made with the part-of-speech (POS) and dependency tags (DT): for

example, in Figure 4, *"NN nsubj *JJ* nmod NN nmod NN"*[2]. The second and the third patterns are built by keeping only either the POS or the DT. The patterns associated with our running example are then: *"NN *JJ* NN NN"* and *"nsubj ** nmod nmod"*. For a given pattern, the *syntactic similarity* score is obtained using the following procedure: Given 2 corpora, (1) we first extract all the shortest path pattern that appear between two related entities. (2) For each corpus, we compute the pattern distribution (*i.e.*, the list of patterns, along with their frequency) by normalizing over all the patterns in the corpus. (3) The score is then computed with the cosine similarity between the pattern distributions of two corpora. Table 4 shows the cosine similarity measures between target and source corpora for the three different pattern distributions. We observe that, for the two target corpora, the performance gain obtained using the TL strategy using a given source corpus can be related to the cosine similarity with this corpus: the higher cosine similarity lead to the best transfer TL.

		Source corpora		
		DDI	**ADE**	**reACE**
		POS + DT		
Source corpora	**SNPPhena**	0.53	0.22	0.13
	EU-ADR	0.24	0.20	0.09
		POS only		
	SNPPhena	0.80	0.70	0.35
	EU-ADR	0.77	0.68	0.32
		DT only		
	SNPPhena	0.53	0.23	0.14
	EU-ADR	0.25	0.24	0.10

Table 4: Cosine similarity score between target and source corpora for the three different pattern distributions. POS is part of speech pattern and DT is dependency type pattern.

Dictionary coverage. On the opposite, we observed that the efficiency of TL in our experiments can not be fully explained by the lexical similarity between source and target corpora. As shown in Table 5, the vocabulary overlap with the target corpora is almost equivalent whether we are considering DDI or ADE (53.4 *vs.* 51.2 and 58.9 *vs.* 60.5), whereas performances obtained with DDI were better than those obtained with ADE. Unsurprisingly, it is lower for reACE which is not a biomedical corpus.

	DDI	ADE	reACE
SNPPhenA	53.4	51.2	39.8
EU-ADR	58.9	60.5	38.3

Table 5: Dictionary coverage. Percentage of words from the target copora present in the source corpora.

Lexical and semantic paradigms. We complete this analysis with few examples illustrating the lexical and semantic heterogeneity of sentences that may instantiate a same pattern. Table 6 provides 4 patterns and their instantiations in source and target corpora. One can observe that sentences instantiating a same pattern seems to have no particular similarity when considering lexical and semantic paradigms. A similar heterogeneity is observed when considering the lowest common ancestor term (or the *head*) of the patterns. Table 7 lists the most frequent lowest common ancestor in each corpus. Again, we observe no direct link with learning improvement.

7 Conclusion

In this paper, we empirically showed that a TL strategy can benefit biomedical RE tasks when using the TreeLSTM model, whereas it is mainly harmful with a model that does not consider syntax. This is of great interest for specific domains, such those of biomedicine, for which few annotated resources are available. Our TL approach led *(i)* to better performances than the state of the art for two biomedical RE tasks: SNP-phenotype and drug-disease RE; and *(ii)* to state-of-the-art results for two others focusing on target-disease and target-drug relationships. Interestingly, we showed that even a general domain corpus (reACE) may carry useful information and lead to improved performances. We proposed an analysis with syntax-based metrics and examples to provide elements of interpretation of this behavior and emphasize the key role of syntax in TL for RE. An exciting direction would be to explore this transfer strategy with Electronic Health Records of various origin.

Aknowledgement

This work is funded by the French National Research Agency (ANR) under the *PractiKPharma* project: ANR-15-CE23-0028, by the IDEX "Lorraine Université d'Excellence" (15-IDEX-0004) and by the *Snowball* Inria Associate Team.

[2]The stars mark the lowest common ancestor of the two entities in the dependency tree and are used to prevent similar pattern with different common ancestors to be considered the same. Note that the patterns are not directed, thus the two patterns *"NN nsubj *JJ* nmod NN nmod NN"* and *"NN nmod NN nmod *JJ* nsubj NN"* are equivalent.

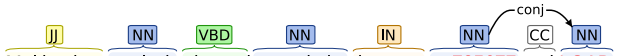
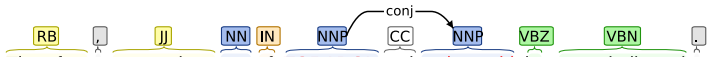
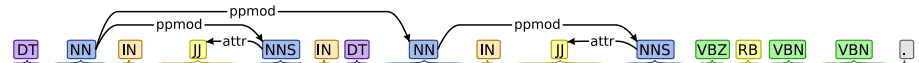
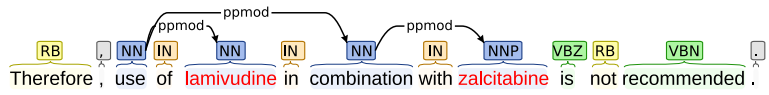
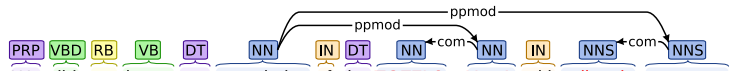
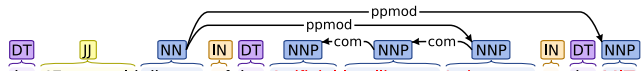
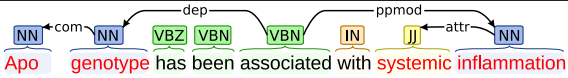
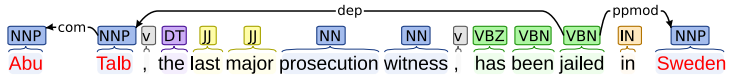
Pattern	Corpus	Example of instantiation
***NN* conj NN**	SNPPhenA	
	DDI	
ppmod *NN* ppmod NN	EU-ADR	
ppmod NN	DDI	
NN pmod *NN* ppmod NN	SNPPhenA	
	reACE	
NN dep *VBN* ppmod NN	SNPPhenA	
	reACE	

Table 6: Examples of patterns and of their instantiation in corpora. Red words correspond to entities.

SNPPhenA	EU-ADR	DDI	ADE	reACE
associated (25.2)	analyzed (5.8)	*entity* (17.8)	*entity* (30.1)	*entity* (60.6)
entity (12.2)	associated (4.3)	administered (4.1)	developed (11.1)	is (2.2)
genotyped (5.4)	*entity* (2.9)	increase (3.0)	associated (4.1)	was (1.9)
association (4.4)	is (2.9)	administration (2.7)	is (2.7)	said (1.4)
showed (3.8)	polymorphisms (2.4)	reported (2.6)	induced (2.3)	
observed (3.3)	over-represented (2.4)	interact (2.6)	case (1.6)	
genes (2.6)	showed (2.4)	reduce (2.5)	following (1.4)	

Table 7: Terms corresponding to the lowest common ancestor in the POS + DT patterns. Their relative frequency in each corpus is provided in parenthesis. *Entity* means that the term is one of the two entities.

References

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175.

Christopher M. Bishop. 2007. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer.

Behrouz Bokharaeian, Alberto Díaz Esteban, Nasrin Taghizadeh, Hamidreza Chitsaz, and Ramyar Chavoshinejad. 2017. SNPPhenA: a corpus for extracting ranked associations of single-nucleotide polymorphisms and phenotypes from literature. *J. Biomedical Semantics*, 8(1):14:1–14:13.

Àlex Bravo, Janet Piñero González, Núria Queralt-Rosinach, Michael Rautschka, and Laura Inés Furlong. 2015. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC Bioinformatics*, 16:55:1–55:17.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, pages 724–731.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Peggy Cellier, Thierry Charnois, and Marc Plantevit. 2010. Sequential patterns to discover and characterise biological relations. In *Computational Linguistics and Intelligent Text Processing, 11th International Conference, CICLing 2010, Iasi, Romania, March 21-27, 2010. Proceedings*, pages 537–548.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Nicolas Fiorini, David J. Lipman, and Zhiyong Lu. 2017. Towards PubMed 2.0. *Elife*, 6.

Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 425–429.

Liang Ge, Jing Gao, Hung Q. Ngo, Kang Li, and Aidong Zhang. 2014. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining*, 7(4):254–271.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE.

Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. 2012. Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics*, 3(1):15.

Ben Hachey, Claire Grover, and Richard Tobin. 2012. Datasets for generic relation extraction. *Natural Language Engineering*, 18(1):21–59.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions. *Journal of Biomedical Informatics*, 46(5):914–920.

Fei Huang and Alexander Yates. 2009. Distributional representations for handling sparsity in supervised sequence-labeling. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 495–503.

Yoon Kim. 2014. 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*, pages 1746–1751.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 482–490.

Joël Legrand and Ronan Collobert. 2014. Joint RNN-based greedy parsing and word composition. *CoRR*, abs/1412.7028.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and WANG Houfeng. 2015. A dependency-based neural network for relation classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 101–104.

Mike Mintz, Steven Bills, Rion Snow, and Dan 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: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

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, ACL 2016*.

Erik M. van Mulligen, Annie Fourrier-Réglat, David Gurwitz, Mariam Molokhia, Ainhoa Nieto, Gianluca Trifirò, Jan A. Kors, and Laura Inés Furlong. 2012. The EU-ADR corpus: Annotated drugs, diseases, targets, and their relationships. *Journal of Biomedical Informatics*, 45(5):879–884.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1498–1507.

Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. 2016. Multichannel Convolutional Neural Network for Biological Relation Extraction. *BioMed research international*, 2016:1850404.

Herbert Robbins and Sutton Monro. 1985. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer.

Chun-Wei Seah, Yew-Soon Ong, and Ivor W. Tsang. 2013. Combating negative transfer from predictive distribution differences. *IEEE Trans. Cybernetics*, 43(4):1153–1165.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013*, pages 455–465.

Nitish Srivastava, Geoffrey E. 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(1):1929–1958.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015*, pages 1556–1566.

Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. 2016. A survey of transfer learning. *J. Big Data*, 3:9.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 536–540.

Yunlun Yang, Yunhai Tong, Shulei Ma, and Zhi-Hong Deng. 2016. A position encoding convolutional neural network based on dependency tree for relation classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 65–74.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING 2014, 25th International Conference on Computational Linguistics*, pages 2335–2344.

Han Zhao, Shanghang Zhang, Guanhang Wu, João P Costeira, José MF Moura, and Geoffrey J Gordon. 2017. Multiple source domain adaptation with adversarial training of neural networks. *arXiv preprint arXiv:1705.09684*.