

# Drug-drug Interaction Extraction Using Composite Kernels

Md. Faisal Mahbub Chowdhury<sup>1,2</sup> and Alberto Lavelli<sup>1</sup>

<sup>1</sup> HLT Research Unit, Fondazione Bruno Kessler (FBK), Trento, Italy

<sup>2</sup> Department of Information Eng. and Computer Science, University of Trento, Italy  
{chowdhury,lavelli}@fbk.eu

**Abstract.** Detection of drug-drug interaction (DDI) is crucial for identification of adverse drug effects. In this paper, we present a range of new composite kernels that are evaluated in the DDIExtraction2011 challenge. These kernels are computed using different combinations of tree and feature based kernels. The best result that we obtained is an  $F_1$  score of 0.6370 which is higher than the already published result on this same corpus.

**Keywords:** drugs, kernels, dependency tree, phrase structure tree, local context, global context.

## 1 Introduction

The *DDIExtraction2011 challenge*<sup>3</sup> provides a platform to identify the state of the art for drug-drug interaction (DDI) extraction from biomedical articles. We have participated in this challenge applying a range of new composite kernels. These kernels combine different combinations of *mildly extended dependency tree (MEDT)* kernel [2], *phrase structure tree (PST)* kernel [7], *local context (LC)* kernel [4], *global context (GC)* kernel [4] and *shallow linguistic (SL)* kernel [4].

The best result we have obtained is an  $F_1$  score of **0.6370** by combining MEDT, PST and GC kernels on the unified format of the data. From the pre-processing of data to the extraction of DDIs using kernel compositions, our objective is to exploit the maximum information that could be learned from different representations of the data.

In the remaining of this paper, we discuss how we have addressed the DDI extraction task. In Section 2, we briefly discuss the dataset. Then in Section 3, the pre-processing steps are described. Following that, in Section 4, we mention the individual kernels which are the building blocks for our kernel compositions. Section 5 defines the proposed composite kernels. Evaluation results are discussed in Section 6. Finally, in Section 7 we summarize our work and present ideas for future work.

---

<sup>3</sup> <http://labda.inf.uc3m.es/DDIExtraction2011/>

## 2 Dataset

The DDIExtraction2011 challenge task requires the automatic identification of DDIs from biomedical articles. Only the intra-sentential DDIs (i.e. DDIs within single sentence boundaries) are considered. The challenge corpus [9] is divided into training and evaluation datasets. Initially released training data consists of 435 abstracts and 4,267 sentences, and is annotated with 2,402 DDIs. During the evaluation phase, a dataset containing 144 abstracts and 1,539 sentences is provided to the participants as the evaluation data. Both the datasets contain drug annotations, but only the training dataset has DDI annotations.

These datasets are made available in two formats: the so-called *unified* format and the *MMTx* format. The unified format contains only the tokenized sentences, while the MMTx format contains the tokenized sentences along with POS tag for each token.

We have used the unified format data. We have found out that, in both training and evaluation datasets, there are some missing special symbols, perhaps due to encoding problems. The position of these symbols can be identified by the presence of the question mark “?” symbol. For example:

```
<sentence id="DrugDDI.d554.s14" origId="s14" text="Ergotamine
or dihydroergotamine?acute ergot toxicity characterized by severe periph-
eral vasospasm and dysesthesia.">
```

We have tried to randomly check whether the unified format and MMTx format datasets contain the same sentences. We have found that one of the randomly chosen sentences<sup>4</sup> does not include a “>” character which exists as a token of the corresponding sentence inside the corresponding MMTx file. This suggests that there might be missing characters inside some sentences due to conversion errors of the html/xml special characters.

## 3 Data pre-processing

Our system is trained and evaluated on the unified format. We use the Stanford parser<sup>5</sup> [6] for tokenization, POS-tagging and parsing of the sentences.

Some of the characteristics of the data sets have required pre-processing steps to correctly handle the texts. Having “?” in the middle of a sentence causes parsing errors since the syntactic parser often misleadingly considers it as a sentence ending sign. So, we replaced all “?” with “@”. Additionally, to reduce tokenization errors, if a drug name does not contain an empty space character immediately before and after its boundaries, we insert space characters in those positions inside the corresponding sentence.

The SPECIALIST lexicon tool is used to normalize tokens to avoid spelling variations and also to provide lemmas. The dependency relations produced by the parser are used to create dependency parse trees for corresponding sentences.

<sup>4</sup> DrugDDI.d151.s11 of the file Flumazenil.ddi.xml.

<sup>5</sup> <http://nlp.stanford.edu/software/lex-parser.shtml>

## 4 Individual kernel approaches that we exploit

The approach adopted for our participation to the challenge is to exploit systems (or methodologies) that already obtained state-of-the-art results in the protein-protein interaction (PPI) extraction task and also in other RE tasks in domains such as newspaper articles. One of these systems [4] uses feature based kernels and is shown to be very effective for PPI extraction. We also consider tree kernel based approaches since they are the state of the art for various RE tasks especially from newspaper texts. All of the systems (or methodologies) are based on the support vector machine (SVM) algorithm for supervised machine learning.

### 4.1 Feature based kernels

Giuliano et al. [4] proposed a so called Shallow Linguistic (SL) kernel which is so far one of the best performing kernels used for biomedical RE. The SL kernel is defined as follows:

$$K_{SL}(R_1, R_2) = K_{LC}(R_1, R_2) + K_{GC}(R_1, R_2)$$

where  $K_{SL}$ ,  $K_{GC}$  and  $K_{LC}$  correspond to SL, global context (GC) and local context (LC) kernels respectively. The GC kernel exploits contextual information of the words occurring before, between and after the pair of entities (to be investigated for RE) in the corresponding sentence; while the LC kernel exploits contextual information surrounding individual entities.

The jsRE system<sup>6</sup> provides an implementation of these kernels. It should be noted that, by default, jsRE uses the ratio of negative and positive examples as the value of the cost-ratio-factor<sup>7</sup> parameter during SVM training.

Segura-Bedmar et al. [9] used the jsRE system for DDI extraction on the same corpus (in the MMTx format) that has been used during the DDIExtraction2011 challenge. They experimented with various parameter settings, and reported an  $F_1$  score of 0.6001. We used the same parameter settings (n-gram=3, window-size=3) with which they obtained their best result.

### 4.2 Tree kernels

One of the tree kernels that we have used is called mildly extended dependency tree (MEDT) kernel, proposed by Chowdhury et al. [2]. A dependency tree (DT) kernel, pioneered by Culotta et al. [3], is typically applied to the minimal or smallest common subtree of a dependency parse tree that includes a target pair of entities. Such subtree reduces unnecessary information by placing word(s) closer to its dependent(s) inside the tree and emphasizes local features of the

<sup>6</sup> <http://hlt.fbk.eu/en/technology/jsRE>

<sup>7</sup> This parameter value is the one by which training errors on positive examples would outweigh errors on negative examples.

corresponding relation. However, sometimes a minimal subtree might not contain important cue words or predicates.

The MEDT kernel addresses this issue using some linguistically motivated extensions. The best settings for the MEDT kernel, that we used in our experiments for DDI extraction, observed by the authors on the AIMed protein-protein interaction dataset [1] is by expanding the minimal subtree with the following rule, and then by using unlexicalized partial trees (uPTs) [10] for similarity matching.

*If the root of the minimal subtree is the head word of one of the interacting entities, then add the parent node (in the original DT tree) of the root node as the new root of the subtree.*

Apart from that, we have also used a phrase structure tree (PST) kernel which is basically the path-enclosed tree (PET) proposed by Moschitti [7]. This tree kernel is based on the smallest common subtree of a phrase structure parse tree, which includes the two entities involved in a relation.

## 5 Proposed kernel compositions

We propose the following composite kernels for DDI extraction:

- $K_{MP}(R_1, R_2) = w_1 * K_{MEDT}(R_1, R_2) + w_2 * K_{PST}(R_1, R_2)$
- $K_{LMP}(R_1, R_2) = K_{LC}(R_1, R_2) + w_3 * K_{MP}$
- $K_{GMP}(R_1, R_2) = K_{GC}(R_1, R_2) + w_3 * K_{MP}$
- $K_{SMP}(R_1, R_2) = K_{SL}(R_1, R_2) + w_3 * K_{MP}$

where  $K_{SL}$ ,  $K_{MEDT}$ ,  $K_{PST}$ ,  $K_{LC}$  and  $K_{GC}$  represent SL, MEDT, PST, LC and GC kernels respectively, and  $w_i$  represents multiplicative constant(s). The values for all of the  $w_i$  used during our experiments are equal to 1. All the composite kernels are valid according to the closure properties of kernels.

The motivation behind using these new composite kernels is to combine varying representations (i.e. tree structures and flat structures) of different types of information (i.e. dependencies, syntactic information, and shallow linguistic features), and to see whether they can complement each other to learn a more robust model.

To compute the feature vectors of  $K_{SL}$ ,  $K_{LC}$  and  $K_{GC}$ , we used the jSRE system. The tree kernels and composite kernels are computed using the SVM-LIGHT-TK toolkit<sup>8</sup> [8, 5]. Finally, the ratio of negative and positive examples has been used as the value of the cost-ratio-factor parameter.

<sup>8</sup> <http://disi.unitn.it/moschitti/Tree-Kernel.htm>

## 6 Evaluation Results

We have tuned all the composite kernels on the training data using 10-fold cross validation<sup>9</sup>. The results of these experiments are shown in Table 1. The experiments show that the best result is gained using the  $K_{GMP}$ . Both the  $K_{GMP}$  and  $K_{SMP}$  perform much better than the other kernels.

	$K_{MP}$	$K_{SL}$	$K_{LMP}$	$K_{GMP}$	$K_{SMP}$
Precision	0.4836	0.5109	0.5150	0.5522	0.5616
Recall	0.6249	0.6607	0.6507	0.6520	0.6336
$F_1$ Score	0.5452	0.5762	0.5750	<b>0.5980</b>	0.5954

**Table 1.** Results of 10-fold cross validation on the training data.

Table 2 shows the official evaluation results of our proposed kernels in the challenge. The results show a trend similar to the one of the cross-validation, with the composite tree kernel  $K_{MP}$  obtaining an  $F_1$  score much lower than that of the other kernels. The combination of the tree and feature based kernels produces better results as the  $K_{SMP}$  got a better  $F_1$  score than that of the  $K_{SL}$  or  $K_{MP}$  alone. But this combination also caused a drop in true and false positives. This suggests that such a combination produces a conservative model that requires more similarities in the features and structures of the candidate relations to be identified as DDIs than that of the kernels they are composed of.

As expected, the highest result is obtained by the  $K_{GMP}$  kernel. Implicitly, the tree kernels already exploit local contextual features as part of the tree structures. For example, the lemma of a (relevant) token is considered as a node inside the MEDT structure, while the order of the neighbouring tokens of an entity (along with their POS tags) are inherited inside the PST structure. So, excluding the LC kernel in the composite kernel might have been allowed to avoid data overfitting. Furthermore, entity blinding (i.e. generalizing the named entities instead of using their original names) is not considered for the basic feature set construction of LC kernel (please refer to the original paper of Giuliano et al. [4]). This might have caused systematic bias and resulted in lower performance.

## 7 Conclusion

In this paper, we have applied new composite kernels that exploit different types of tree structures and features. These include dependency trees and phrase structure trees as well as local and global contexts of the relevant entities. The kernels

<sup>9</sup> To obtain the overall performance we sum up the true positives, false positives, and false negatives of all the 10 folds, and then measure precision, recall and  $F_1$  score from these figures.

	$K_{MP}$	$K_{SL}$	$K_{LMP}$	$K_{GMP}$	$K_{SMP}$
True Positive	544	560	534	529	513
False Positive	674	458	423	377	344
False Negative	211	195	221	226	242
True Negative	5597	5813	5848	5894	5927
Precision	0.4466	0.5501	0.558	0.5839	0.5986
Recall	0.7205	0.7417	0.7073	0.7007	0.6795
$F_1$ Score	0.5514	0.6317	0.6238	<b>0.6370</b>	0.6365

**Table 2.** Evaluation results on the test data provided by the challenge organisers.

have been evaluated on the DDIExtraction2011 challenge, and have achieved encouraging results.

Due to time constraints, we have not been able to perform extensive parameter tuning. We are confident that tuning of the multiplicative constant(s) (i.e.  $w_i$ ) might produce even better performance. We also predict these kernels would be able to learn more accurate training models using a bigger training data, and would produce results better than that of the individual kernels which are their building blocks.

## Acknowledgments

This work was carried out in the context of the project “eOnco - Pervasive knowledge and data management in cancer care”.

## References

- [1] Bunescu, R., Ge, R., Kate, R.J., Marcotte, E.M., Mooney, R.J., Ramani, A.K., Wong, Y.W.: Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine (Special Issue on Summarization and Information Extraction from Medical Documents)* 33(2), 139–155 (2005)
- [2] Chowdhury, M.F.M., Lavelli, A., Moschitti, A.: A study on dependency tree kernels for automatic extraction of protein-protein interaction. In: *Proceedings of BioNLP 2011 Workshop*. pp. 124–133. Association for Computational Linguistics, Portland, Oregon, USA (June 2011)
- [3] Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona, Spain (2004)
- [4] Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL’2006)*. pp. 401–408. Trento, Italy (2006)
- [5] Joachims, T.: Making large-scale support vector machine learning practical. In: *Advances in kernel methods: support vector learning*, pp. 169–184. MIT Press, Cambridge, MA, USA (1999)

- [6] Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL '03). pp. 423–430. Association for Computational Linguistics, Sapporo, Japan (2003)
- [7] Moschitti, A.: A study on convolution kernels for shallow semantic parsing. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. ACL '04, Barcelona, Spain (2004)
- [8] Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) Machine Learning: ECML 2006, Lecture Notes in Computer Science, vol. 4212, pp. 318–329. Springer Berlin / Heidelberg (2006)
- [9] Segura-Bedmar, I., Martínez, P., Pablo-Sánchez, C.d.: Using a shallow linguistic kernel for drug-drug interaction extraction. *Journal of Biomedical Informatics* In Press, Corrected Proof, Available online (24 April, 2011)
- [10] Severyn, A., Moschitti, A.: Fast cutting plane training for structural kernels. In: Proceedings of ECML-PKDD (2010)