Combining Tree Structures, Flat Features and Patterns for Biomedical Relation Extraction

Md. Faisal Mahbub Chowdhury [†] [‡] **and Alberto Lavelli** [‡]
[‡] Fondazione Bruno Kessler (FBK-irst), Italy
[†] University of Trento, Italy
{chowdhury,lavelli}@fbk.eu

Abstract

Kernel based methods dominate the current trend for various relation extraction tasks including protein-protein interaction (PPI) extraction. PPI information is critical in understanding biological processes. Despite considerable efforts, previously reported PPI extraction results show that none of the approaches already known in the literature is consistently better than other approaches when evaluated on different benchmark PPI corpora. In this paper, we propose a novel hybrid kernel that combines (automatically collected) dependency patterns, trigger words, negative cues, walk features and regular expression patterns along with tree kernel and shallow linguistic kernel. The proposed kernel outperforms the exiting state-of-the-art approaches on the BioInfer corpus, the largest PPI benchmark corpus available. On the other four smaller benchmark corpora, it performs either better or almost as good as the existing approaches. Moreover, empirical results show that the proposed hybrid kernel attains considerably higher precision than the existing approaches, which indicates its capability of learning more accurate models. This also demonstrates that the different types of information that we use are able to complement each other for relation extraction.

1 Introduction

Kernel methods are considered the most effective techniques for various relation extraction (RE) tasks on both general (e.g. newspaper text) and specialized (e.g. biomedical text) domains. In particular, as the importance of syntactic structures for deriving the relationships between entities in text has been growing, several graph and tree kernels have been designed and experimented.

Early RE approaches more or less fall in one of the following categories: *(i)* exploitation of statistics about co-occurrences of entities, *(ii)* usage of patterns and rules, and *(iii)* usage of flat features to train machine learning (ML) classifiers. These approaches have been studied for a long period and have their own pros and cons. Exploitation of co-occurrence statistics results in high recall but low precision, while rule or pattern based approaches can increase precision but suffer from low recall. Flat feature based ML approaches employ various kinds of linguistic, syntactic or contextual information and integrate them into the feature space. They obtain relatively good results but are hindered by drawbacks of limited feature space and excessive feature engineering. Kernel based approaches have become an attractive alternative solution, as they can exploit huge amount of features without an explicit representation.

In this paper, we propose a new hybrid kernel for RE. We apply the kernel to Protein–protein interaction (PPI) extraction, the most widely researched topic in biomedical relation extraction. PPI[1] information is very critical in understanding biological processes. Considerable progress has been made for this task. Nevertheless, empirical results of previous studies show that none of the approaches already known in the literature is consistently better than other approaches when evaluated on different benchmark PPI corpora (see Table 4). This demands further study and innovation

[1] PPIs occur when two or more proteins bind together, and are integral to virtually all cellular processes, such as metabolism, signalling, regulation, and proliferation (Tikk et al., 2010).

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 420–429,
Avignon, France, April 23 - 27 2012. ©2012 Association for Computational Linguistics

of new approaches that are sensitive to the variations of complex linguistic constructions.

The proposed hybrid kernel is the composition of one tree kernel and two feature based kernels (one of them is already known in the literature and the other is proposed in this paper for the first time). The novelty of the newly proposed feature based kernel is that it envisages to accommodate the advantages of pattern based approaches. More precisely:

1. We propose a new feature based kernel (details in Section 4.1) by using syntactic dependency patterns, trigger words, negative cues, regular expression (henceforth, regex) patterns and walk features (i.e. e-walks and v-walks)[2].

2. The syntactic dependency patterns are automatically collected from a type of dependency subgraph (we call it *reduced graph*, more details in Section 4.1.1) during runtime.

3. We only use the regex patterns, trigger words and negative cues mentioned in the literature (Ono et al., 2001; Fundel et al., 2007; Bui et al., 2010). The objective is to verify whether we can exploit knowledge which is already known and used.

4. We propose a hybrid kernel by combining the proposed feature based kernel (outlined above) with the Shallow Linguistic (SL) kernel (Giuliano et al., 2006) and the Path-enclosed Tree (PET) kernel (Moschitti, 2004).

The aim of our work is to take advantage of different types of information (i.e., dependency patterns, regex patterns, trigger words, negative cues, syntactic dependencies among words and constituent parse trees) and their different representations (i.e. flat features, tree structures and graphs) which can complement each other to learn more accurate models.

The remainder of the paper is organized as follows. In Section 2, we briefly review previous work. Section 3 lists the datasets. Then, in Section 4, we define our proposed hybrid kernel and describe its individual component kernels. Section 5 outlines the experimental settings. Following that, empirical results are discussed in Section 6. Finally, we conclude with a summary of our study as well as suggestions for further improvement of our approach.

2 Related Work

In this section, we briefly discuss some of the recent work on PPI extraction. Several RE approaches have been reported to date for the PPI task, most of which are kernel based methods. Tikk et al. (2010) reported a benchmark evaluation of various kernels on PPI extraction. An interesting finding is that the Shallow Linguistic (SL) kernel (Giuliano et al., 2006) (to be discussed in Section 4.2), despite its simplicity, is on par with the best kernels in most of the evaluation settings.

Kim et al. (2010) proposed walk-weighted subsequence kernel using e-walks, partial matches, non-contiguous paths, and different weights for different sub-structures (which are used to capture structural similarities during kernel computation). Miwa et al. (2009a) proposed a hybrid kernel, which combines the all-paths graph (APG) kernel (Airola et al., 2008), the bag-of-words kernel, and the subset tree kernel (Moschitti, 2006) (applied on the shortest dependency paths between target protein pairs). They used multiple parser inputs. The system is regarded as the current state-of-the-art PPI extraction system because of its high results on different PPI corpora (see the results in Table 4).

As an extension of their work, they boosted system performance by training on multiple PPI corpora instead of on a single corpus and adopting a corpus weighting concept with support vector machine (SVM) which they call SVM-CW (Miwa et al., 2009b). Since most of their results are reported by training on the combination of multiple corpora, it is not possible to compare them directly with the results published in the other related works (that usually adopt 10-fold cross validation on a single PPI corpus). To be comparable with the vast majority of the existing work, we also report results using 10-fold cross validation

[2] The syntactic dependencies of the words of a sentence create a dependency graph. A **v-walk** feature consists of $(word_i - dependency\_type_{i,i+1} - word_{i+1})$, and an **e-walk** feature is composed of $(dependency\_type_{i-1,i} - word_i - dependency\_type_{i,i+1})$. Note that, in a dependency graph, the words are nodes while the dependency types are edges.

Corpus	Sentences	Positive pairs	Negative pairs
BioInfer	1,100	2,534	7,132
AIMed	1,955	1,000	4,834
IEPA	486	335	482
HPRD50	145	163	270
LLL	77	164	166

Table 1: Basic statistics of the 5 benchmark PPI corpora.

on single corpora.

Apart from the approaches described above, there also exist other studies that used kernels for PPI extraction (e.g. subsequence kernel (Bunescu and Mooney, 2006)).

A notable exception is the work published by Bui et al. (2010). They proposed an approach that consists of two phases. In the first phase, their system categorizes the data into different groups (i.e. subsets) based on various properties and patterns. Later they classify candidate PPI pairs inside each of the groups using SVM trained with features specific for the corresponding group.

3 Data

There are 5 benchmark corpora for the PPI task that are frequently used: HPRD50 (Fundel et al., 2007), IEPA (Ding et al., 2002), LLL (Nédellec, 2005), BioInfer (Pyysalo et al., 2007) and AIMed (Bunescu et al., 2005). These corpora adopt different PPI annotation formats. For a comparative evaluation Pyysalo et al. (2008) put all of them in a common format which has become the standard evaluation format for the PPI task. In our experiments, we use the versions of the corpora converted to such format.

Table 1 shows various statistics regarding the 5 (converted) corpora.

4 Proposed Hybrid Kernel

The hybrid kernel that we propose is as follows:

$$K_{Hybrid}\ (R_1, R_2) = K_{TPWF}\ (R_1, R_2) \\ + K_{SL}\ (R_1, R_2) + w * K_{PET}\ (R_1, R_2)$$

where K_{TPWF} stands for the new feature based kernel (henceforth, TPWF kernel) computed using flat features collected by exploiting patterns, trigger words, negative cues and walk features. K_{SL} and K_{PET} stand for the Shallow Linguistic (SL) kernel and the Path-enclosed Tree (PET) kernel respectively. w is a multiplicative constant used for the PET kernel. It allows the hybrid kernel to assign more (or less) weight to the information obtained using tree structures depending on the corpus. The proposed hybrid kernel is valid according to the closure properties of kernels.

Both the TPWF and SL kernels are linear kernels, while PET kernel is computed using Unlexicalized Partial Tree (uPT) kernel (Severyn and Moschitti, 2010). The following subsections explain each of the individual kernels in more detail.

4.1 Proposed TPWF Kernel

4.1.1 Reduced graph, trigger words, negative cues and dependency patterns

For each of the candidate entity pairs, we construct a type of subgraph from the dependency graph formed by the syntactic dependencies among the words of a sentence. We call it "`reduced graph`" and define it in the following way:

> A **reduced graph** is a subgraph of the dependency graph of a sentence which includes:
>
> - the two candidate entities and their governor nodes up to their least common governor (if exists).
> - dependent nodes (if exist) of all the nodes added in the previous step.
> - the immediate governor(s) (if exists) of the least common governor.

Figure 1 shows an example of a reduced graph. A reduced graph is an extension of the smallest common subgraph of the dependency graph that aims at overcoming its limitations. It is a known issue that the smallest common subgraph (or subtree) sometimes does not contain cue words. Previously, Chowdhury et al. (2011a) proposed a linguistically motivated extension of the minimal (i.e. smallest) common subtree (which includes the candidate entity pairs), known as Mildly Extended Dependency Tree (MEDT). However, the rules used for MEDT are too constrained. Our objective in constructing the reduced graph is *to include any potential modifier(s) or cue word(s)* that describes the relation between the given pair of entities. Sometimes such modifiers or cue words are not directly dependent (syntactically) on any

	BioInfer			AIMed			IEPA			HPRD50			LLL		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Only walk features	51.8	71.2	60.0	48.7	63.2	55.0	61.0	75.2	67.4	60.2	65.0	62.5	64.6	87.8	74.4
Features: dep. patterns, trigger, neg. cues, walks	53.8	68.8	60.4	50.6	63.9	56.5	63.9	74.6	68.9	65.0	71.8	68.2	66.5	89.6	76.4
Features: dep. patterns, trigger, neg. cues, walks, regex patterns	53.5	68.6	60.1	52.5	62.9	57.2	63.8	74.6	68.8	65.1	69.9	67.5	67.4	88.4	76.5

Table 2: Results of the proposed TPWF feature based kernel on 5 benchmark PPI corpora before and after adding features collected using dependency patterns, regex patterns, trigger words and negative cues to the walk features. The TPWF kernel is a component of the new hybrid kernel.

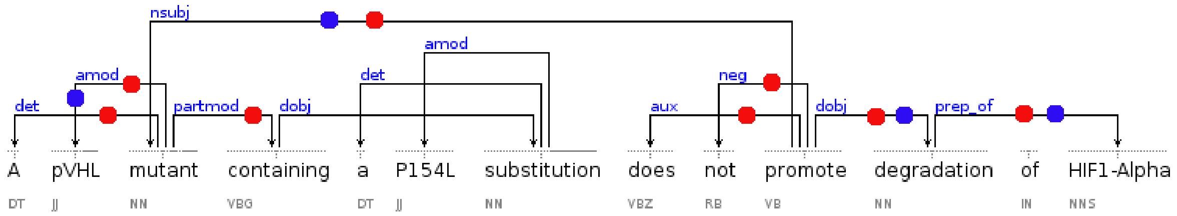

Figure 1: Dependency graph for the sentence "A pVHL mutant containing a P154L substitution does not promote degradation of HIF1-Alpha" generated by the Stanford parser. The edges with blue dots form the smallest common subgraph for the candidate entity pair **pVHL** and **HIF1-Alpha**, while the edges with red dots form the *reduced graph* for the pair.

of the entities (of the candidate pair). Rather they are dependent on some other word(s) which is dependent on one (or both) of the entities. The word "*not*" in Figure 1 is one such example. The reduced graph aims to preserve these cue words.

The following types of features are collected from the reduced graph of a candidate pair:

1. *HasTriggerWord*: whether the least common governor(s) of the target entity pairs inside the reduced graph matches any trigger word.

2. *Trigger-X*: whether the least common governor(s) of the target entity pairs inside the reduced graph matches the trigger word 'X'.

3. *HasNegWord*: whether the reduced graph contains any negative word.

4. *DepPattern-i*: whether the reduced graph contains all the syntactic dependencies of the *i*-th pattern of dependency pattern list.

The dependency pattern list is automatically constructed from the training data during the learning phase. Each pattern is a set of syntactic dependencies of the corresponding reduced graph of a (positive or negative) entity pair in the training data. For example, the dependency pattern for the reduced graph in Figure 1 is {*det, amod, partmod, nsubj, aux, neg, dobj, prep_of*}. The same dependency pattern might be constructed for multiple (positive or negative) entity pairs. However, if it is constructed for both positive and negative pairs, it has to be discarded from the pattern list.

The dependency patterns allow some kind of underspecification as they do not contain the lexical items (i.e. words) but contain the likely combination of syntactic dependencies that a given related pair of entities would pose inside their reduced graph.

The list of trigger words contains 144 words previously used by Bui et al. (2010) and Fundel et al. (2007). The list of negative cues contain 18 words, most of which are mentioned in Fundel et al. (2007).

4.1.2 Walk features

We extract *e-walk* and *v-walk* features from the Mildly Extended Dependency Tree (MEDT) (Chowdhury et al., 2011a) of each candidate pair. Reduced graphs sometimes include some unin-

	BioInfer			**AIMed**			**IEPA**			**HPRD50**			**LLL**		
Pos. / Neg.	2,534 / 7,132			1,000 / 4,834			335 / 482			163 / 270			164 / 166		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Proposed TPWF kernel (without regex)	53.8	68.8	60.4	50.6	63.9	56.5	63.9	74.6	68.9	65.0	71.8	68.2	66.5	89.6	76.4
Proposed TPWF kernel (with regex)	53.5	68.6	60.1	52.5	62.9	57.2	63.8	74.6	68.8	65.1	69.9	67.5	67.4	88.4	76.5
SL kernel	60.8	65.8	63.2	56.2	64.4	60.0	73.3	71.9	72.6	62.0	65.0	63.5	74.9	85.4	79.8
PET kernel	72.8	74.9	73.9	44.8	72.8	55.5	70.7	77.9	74.2	65.0	73.0	68.8	72.1	89.6	79.9
Proposed hybrid kernel (**PET + SL + TPWF** (without regex))	80.0	71.4	75.5	64.2	58.2	61.1	81.1	69.3	74.7	72.9	59.5	65.5	70.4	95.7	81.1
Proposed hybrid kernel (**PET + SL + TPWF** (with regex))	80.1	72.0	75.9	64.4	58.3	61.2	79.3	69.6	74.1	71.9	61.4	66.2	70.6	95.1	81.0

Table 3: Results of the proposed hybrid kernel and its individual components. *Pos.* and *Neg.* refer to number positive and negative relations respectively. PET refers to the path-enclosed tree kernel, SL refers to the shallow linguistic kernel, and TPWF refers to the kernel computed using trigger, pattern, negative cue and walk features.

formative words which produce uninformative walk features. Hence, they are not suitable for walk feature generation. MEDT suits better for this purpose. The walk features extracted from MEDTs have the following properties:

- The directionality of the edges (or nodes) in an e-walk (or v-walk) is not considered. In other words, e.g., $pos(stimulatory) - amod - pos(effects)$ and $pos(effects) - amod - pos(stimulatory)$ are treated as the same feature.

- The v-walk features are of the form $(pos_i - dependency\_type_{i,i+1} - pos_{i+1})$. Here, pos_i is the POS tag of $word_i$, i is the governor node and $i + 1$ is the dependent node.

- The e-walk features are of the form $(dep.\_type_{i-1,i} - pos_i - dep.\_type_{i,i+1})$ and $(dep.\_type_{i-1,i} - lemma_i - dep.\_type_{i,i+1})$. Here, $lemma_i$ is the lemmatized form of $word_i$.

- Usually, the e-walk features are constructed using dependency types between $\{governor\_of\_X,\ node\_X\}$ and $\{node\_X,\ dependent\_of\_X\}$. However, we also extract e-walk features from the dependency types between any two dependents and their common governor (i.e. $\{node\_X,\ dependent\_1\_of\_X\}$ and $\{node\_X,\ dependent\_2\_of\_X\}$).

Apart from the above types of features, we also add features for lemmas of the immediate preceding and following words of the candidate entities. These feature names are augmented with *-1* or *+1* depending on whether the corresponding words are preceded or followed by a candidate entity.

4.1.3 Regular expression patterns

We use a set of 22 regex patterns as binary features. These patterns were previously used by Ono et al. (2001) and Bui et al. (2010). If there is a match for a pattern (e.g. "*Entity_1.*activates.*Entity_2*" where *Entity_1* and *Entity_2* form the candidate entity pair) in a given sentence, value *1* is added for the feature (i.e., pattern) inside the feature vector.

4.2 Shallow Linguistic (SL) Kernel

The Shallow Linguistic (SL) kernel was proposed by Giuliano et al. (2006). It is one of the best performing kernels applied on different biomedical RE tasks such as PPI and DDI (drug-drug interaction) extraction (Tikk et al., 2010; Segura-Bedmar et al., 2011; Chowdhury and Lavelli, 2011b; Chowdhury et al., 2011c). It is defined as follows:

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

	BioInfer			**AIMed**			**IEPA**			**HPRD50**			**LLL**		
Pos. / Neg.	2,534 / 7,132			1,000 / 4,834			335 / 482			163 / 270			164 / 166		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SL kernel (Giuliano et al., 2006)	—	—	—	60.9	57.2	59.0	—	—	—	—	—	—	—	—	—
APG kernel (Airola et al., 2008)	56.7	67.2	61.3	52.9	61.8	56.4	69.6	82.7	75.1	64.3	65.8	63.4	72.5	87.2	76.8
Hybrid kernel and multiple parser input (Miwa et al., 2009a)	65.7	71.1	68.1	55.0	68.8	60.8	67.5	78.6	71.7	68.5	76.1	70.9	77.6	86.0	80.1
SVM-CW, multiple parser input and graph, walk and BOW features (Miwa et al., 2009b)	–	–	67.6	–	–	64.2	—	—	74.4	—	—	69.7	—	—	80.5
kBSPS kernel (Tikk et al., 2010)	49.9	61.8	55.1	50.1	41.4	44.6	58.8	89.7	70.5	62.2	87.1	71.0	69.3	93.2	78.1
Walk weighted subsequence kernel (Kim et al., 2010)	61.8	54.2	57.6	61.4	53.3	56.6	73.8	71.8	72.9	66.7	69.2	67.8	76.9	91.2	82.4
2 phase extraction (Bui et al., 2010)	61.7	57.5	60.0	55.3	68.5	61.2	—	—	—	—	—	—	—	—	—
Our proposed hybrid kernel (**PET + SL + TPWF** without regex)	80.0	71.4	**75.5**	64.2	58.2	**61.1**	81.1	69.3	**74.7**	72.9	59.5	**65.5**	70.4	95.7	**81.1**

Table 4: Comparison of the results on the 5 benchmark PPI corpora. *Pos.* and *Neg.* refer to number positive and negative relations respectively. The underlined numbers indicate the best results for the corresponding corpus reported by any of the existing state-of-the-art approaches. The results of Bui et al. (2010) on LLL, HPRD50, and IEPA are not reported since thy did not use all the positive and negative examples during cross validation. Miwa et al. (2009b) showed that better results can be obtained using multiple corpora for training. However, we consider only those results of their experiments where they used single training corpus as it is the standard evaluation approach adopted by all the other studies on PPI extraction for comparing results. All the results of the previous approaches reported in this table are directly quoted from their respective original papers.

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.

4.3 Path-enclosed tree (PET) Kernel

The path-enclosed tree (PET) kernel[3] was first proposed by Moschitti (2004) for semantic role labeling. It was later successfully adapted by Zhang et al. (2005) and other works for relation extraction on general texts (such as newspaper domain). A PET is the smallest common subtree of a phrase structure tree that includes the two entities involved in a relation.

A tree kernel calculates the similarity between two input trees by counting the number of common sub-structures. Different techniques have been proposed to measure such similarity. We use the Unlexicalized Partial Tree (uPT) kernel (Severyn and Moschitti, 2010) for the computation of the PET kernel since a comparative evaluation by Chowdhury et al. (2011a) reported that uPT kernels achieve better results for PPI extraction than the other techniques used for tree kernel computation.

[3] Also known as shortest path-enclosed tree (SPT) kernel.

5 Experimental Settings

We have followed the same criteria commonly used for the PPI extraction tasks, i.e. abstract-wise 10-fold cross validation on individual corpus and one-answer-per-occurrence criterion. In fact, we have used exactly the same (abstract-wise) fold splitting of the 5 benchmark (converted) corpora used by Tikk et al. (2010) for benchmarking various kernel methods[4].

The Charniak-Johnson reranking parser (Charniak and Johnson, 2005), along with a self-trained biomedical parsing model (McClosky, 2010), has been used for tokenization, POS-tagging and parsing of the sentences. Before parsing the sentences, all the entities are blinded by assigning names as $EntityX$ where X is the entity index. In each example, the POS tags of the two candidate entities are changed to $EntityX$. The parse trees produced by the Charniak-Johnson reranking parser are then processed by the Stanford parser[5] (Klein and Manning, 2003) to obtain syntactic dependencies according to the Stanford Typed Dependency format.

The Stanford parser often skips some syntactic dependencies in output. We use the following two rules to add some of such dependencies:

- If there is a *"conj_and"* or *"conj_or"* dependency between two words *X* and *Y*, then *X* should be dependent on any word *Z* on which *Y* is dependent and vice versa.

- If there are two verbs *X* and *Y* such that inside the corresponding sentence they have only the word *"and"* or *"or"* between them, then any word *Z* dependent on *X* should be also dependent on *Y* and vice versa.

Our system exploits SVM-LIGHT-TK[6] (Moschitti, 2006; Joachims, 1999). We made minor changes in the toolkit to compute the proposed hybrid kernel. The ratio of negative and positive examples has been used as the value of the cost-ratio-factor parameter. We have done parameter tuning following the approach described by Hsu et al. (2003).

6 Results and Discussion

To measure the contribution of the features collected from the reduced graphs (using dependency patterns, trigger words and negative cues) and regex patterns, we have applied the new TPWF kernel on the 5 PPI corpora before and after using these features. Results shown in Table 2 clearly indicate that usage of these features improve the performance. The improvement of performance is primarily due to the usage of dependency patterns which resulted in higher precision for all the corpora.

We have tried to measure the contribution of the regex patterns. However, from the empirical results a clear trend does not emerge (see Table 2).

Table 3 shows a comparison among the results of the proposed hybrid kernel and its individual components. As we can see, the overall results of the hybrid kernel (with and without using regex pattern features) are better than those by any of its individual component kernels. Interestingly, precision achieved on the 4 benchmark corpora (other than the smallest corpus LLL) is much higher for the hybrid kernel than for the individual components. This strongly indicates that these different types of information (i.e. dependency patterns, regex patterns, triggers, negative cues, syntactic dependencies among words and constituent parse trees) and their different representations (i.e. flat features, tree structures and graphs) can complement each other to learn more accurate models.

Table 4 shows a comparison of the PPI extraction results of our proposed hybrid kernel with those of other state-of-the-art approaches. Since the contribution of regex patterns in the performance of the hybrid kernel was not relevant (as Tables 2 and 3 show), we used the results of proposed hybrid kernel without regex for the comparison. As we can see, the proposed kernel achieves *significantly higher results* on the BioInfer corpus, the largest benchmark PPI corpus (2,534 positive PPI pair annotations) available, than any of the existing approaches. Moreover, the results of the proposed hybrid kernel are on par with the state-of-the-art results on the other smaller corpora.

Furthermore, empirical results show that the proposed hybrid kernel attains *considerably higher precision* than the existing approaches.

[4] Downloaded from http://informatik.hu-berlin.de/forschung /gebiete/wbi/ppi-benchmark .

[5] http://nlp.stanford.edu/software/lex-parser.shtml

[6] http://disi.unitn.it/moschitti/Tree-Kernel.htm

Since a dependency pattern, by construction, contains all the syntactic dependencies inside the corresponding reduced graph, it may happen that some of the dependencies (e.g. *det* or determiner) are not informative for classifying the label of the corresponding class label (i.e., positive or negative relation) of the pattern. Their presence inside a pattern might make it unnecessarily rigid and less general. So, we tried to identify and discard such non informative dependencies by measuring probabilities of the dependencies with respect to the class label and then removing any of them which has probability lower than a threshold (we tried with different threshold values). But doing so decreased the performance. This suggests that the syntactic dependencies of a dependency pattern are not independent of each other even if some of them might have low probability (with respect to the class label) individually. We plan to further investigate whether there could be different criteria for identifying non informative dependencies. For the work reported in this paper, we used the dependency patterns as they are initially constructed.

We also did experiments to see whether collecting features for trigger words from the whole reduced graph would help. But that also decreased performance. This suggests that trigger words are more likely to appear in the least common governors.

7 Conclusion

In this paper, we have proposed a new hybrid kernel for RE that combines two vector based kernels and a tree kernel. The proposed kernel outperforms any of the exiting approaches by a wide margin on the BioInfer corpus, the largest PPI benchmark corpus available. On the other four smaller benchmark corpora, it performs either better or almost as good as the existing state-of-the art approaches.

We have also proposed a novel feature based kernel, called TPWF kernel, using (automatically collected) dependency patterns, trigger words, negative cues, walk features and regular expression patterns. The TPWF kernel is used as a component of the new hybrid kernel.

Empirical results show that the proposed hybrid kernel achieves considerably higher precision than the existing approaches, which indicates its capability of learning more accurate models. This also demonstrates that the different types of information that we use are able to complement each other for relation extraction.

We believe there are at least three ways to further improve the proposed approach. First of all, the 22 regular expression patterns (collected from Ono et al. (2001) and Bui et al. (2010)) are applied at the level of the sentences and this sometimes produces unwanted matches. For example, consider the sentence "$X\ activates\ Y\ and\ inhibits\ Z$" where *X, Y,* and Z are entities. The pattern "$Entity1. * activates. * Entity2$" matches both the *X–Y* and *X–Z* pairs in the sentence. But only the *X–Y* pair should be considered. So, the patterns should be constrained to reduce the number of unwanted matches. For example, they could be applied on smaller linguistic units than full sentences. Secondly, different techniques could be used to identify less-informative syntactic dependencies inside dependency patterns to make them more accurate and effective. Thirdly, usage of automatically collected paraphrases of regular expression patterns instead of the patterns directly could be also helpful. Weakly supervised collection of paraphrases for RE has been already investigated (e.g. Romano et al. (2006)) and, hence, can be tried for improving the TPWF kernel (which is a component of the proposed hybrid kernel).

Acknowledgments

This work was carried out in the context of the project "eOnco - Pervasive knowledge and data management in cancer care". The authors are grateful to Alessandro Moschitti for his help in the use of SVM-LIGHT-TK. We also thank the anonymous reviewers for helpful suggestions.

References

Antti Airola, Sampo Pyysalo, Jari Bjorne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9(Suppl 11):S2.

Quoc-Chinh Bui, Sophia Katrenko, and Peter M.A. Sloot. 2010. A hybrid approach to extract protein-protein interactions. *Bioinformatics*.

Razvan Bunescu and Raymond J. Mooney. 2006. Subsequence kernels for relation extraction. In *Proceedings of NIPS 2006*, pages 171–178.

Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL 2005*.

Md. Faisal Mahbub Chowdhury and Alberto Lavelli. 2011b. Drug-drug interaction extraction using composite kernels. In *Proceedings of DDIExtraction2011: First Challenge Task: Drug-Drug Interaction Extraction*, pages 27–33, Huelva, Spain, September.

Md. Faisal Mahbub Chowdhury, Alberto Lavelli, and Alessandro Moschitti. 2011a. A study on dependency tree kernels for automatic extraction of protein-protein interaction. In *Proceedings of BioNLP 2011 Workshop*, pages 124–133, Portland, Oregon, USA, June.

Md. Faisal Mahbub Chowdhury, Asma Ben Abacha, Alberto Lavelli, and Pierre Zweigenbaum. 2011c. Two dierent machine learning techniques for drug-drug interaction extraction. In *Proceedings of DDIExtraction2011: First Challenge Task: Drug-Drug Interaction Extraction*, pages 19–26, Huelva, Spain, September.

J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. 2002. Mining MEDLINE: abstracts, sentences, or phrases? *Pacific Symposium on Biocomputing*, pages 326–337.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex–relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of EACL 2006*, pages 401–408.

CW Hsu, CC Chang, and CJ Lin, 2003. *A practical guide to support vector classification*. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In *Advances in kernel methods: support vector learning*, pages 169–184. MIT Press, Cambridge, MA, USA.

Seonho Kim, Juntae Yoon, Jihoon Yang, and Seog Park. 2010. Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC Bioinformatics*, 11(1).

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL 2003*, pages 423–430, Sapporo, Japan.

David McClosky. 2010. *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. Ph.D. thesis, Department of Computer Science, Brown University.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2009a. Protein-protein interaction extraction by leveraging multiple kernels and parsers. *International Journal of Medical Informatics*, 78.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2009b. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *Proceedings of EMNLP 2009*, pages 121–130, Singapore.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of ACL 2004*, Barcelona, Spain.

Alessandro Moschitti. 2006. Making Tree Kernels Practical for Natural Language Learning. In *Proceedings of EACL 2006*, Trento, Italy.

Claire Nédellec. 2005. Learning language in logic - genic interaction extraction challenge. *Proceedings of the ICML 2005 workshop: Learning Language in Logic (LLL05)*, pages 31–37.

Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. 2001. Automated extraction of information on protein–protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. 2007. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):50.

Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3):S6.

Lorenza Romano, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. 2006. Investigating a generic paraphrase–based approach for relation extraction. In *Proceedings of EACL 2006*, pages 409–416.

Isabel Segura-Bedmar, Paloma Martínez, and Cesar de Pablo-Sánchez. 2011. Using a shallow linguistic kernel for drug-drug interaction extraction. *Journal of Biomedical Informatics*, In Press, Corrected Proof, Available online, 24 April.

Aliaksei Severyn and Alessandro Moschitti. 2010. Fast cutting plane training for structural kernels. In *Proceedings of ECML-PKDD 2010*.

Domonkos Tikk, Philippe Thomas, Peter Palaga, Jörg Hakenberg, and Ulf Leser. 2010. A Comprehensive Benchmark of Kernel Methods to Extract Protein-Protein Interactions from Literature. *PLoS Computational Biology*, 6(7), July.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Natural Language Processing – IJCNLP 2005*, volume 3651 of *Lecture Notes in Computer Science*, pages 378–389. Springer Berlin / Heidelberg.