A Study on Dependency Tree Kernels for Automatic Extraction of Protein-Protein Interaction

Md. Faisal Mahbub Chowdhury [†‡] and **Alberto Lavelli** [‡] and **Alessandro Moschitti** [†]
[†] Department of Information Engineering and Computer Science, University of Trento, Italy
[‡] Human Language Technology Research Unit, Fondazione Bruno Kessler, Trento, Italy
{chowdhury,lavelli}@fbk.eu, moschitti@disi.unitn.it

Abstract

Kernel methods are considered the most effective techniques for various relation extraction (RE) tasks as they provide higher accuracy than other approaches. In this paper, we introduce new dependency tree (DT) kernels for RE by improving on previously proposed dependency tree structures. These are further enhanced to design more effective approaches that we call mildly extended dependency tree (MEDT) kernels. The empirical results on the protein-protein interaction (PPI) extraction task on the AIMed corpus show that tree kernels based on our proposed DT structures achieve higher accuracy than previously proposed DT and phrase structure tree (PST) kernels.

1 Introduction

Relation extraction (RE) aims at identifying instances of pre-defined relation types in text as for example the extraction of protein-protein interaction (PPI) from the following sentence:

> "Native *C8* also formed a heterodimer with *C5*, and low concentrations of polyionic ligands such as protamine and suramin inhibited the interaction."

After identification of the relevant named entities (NE, in this case *proteins*) *C8* and *C5*, the RE task determines whether there is a PPI relationship between the entities above (which is *true* in the example).

Kernel based approaches for RE have drawn a lot of interest in recent years since they can exploit a huge amount of features without an explicit representation. Some of these approaches are structure kernels (e.g. tree kernels), which carry out structural similarities between instances of relations, represented as phrase structures or dependency trees, in terms of common substructures. Other kernels simply use techniques such as bag-of-words, subsequences, etc. to map the syntactic and contextual information to flat features, and later compute similarity.

One variation of tree kernels is the dependency tree (DT) kernel (Culotta and Sorensen, 2004; Nguyen et al., 2009). A DT kernel (DTK) is a tree kernel that is computed on a dependency tree (or subtree). A dependency tree encodes grammatical relations between words in a sentence where the words are nodes, and dependency types (i.e. grammatical functions of children nodes with respect to their parents) are edges. The main advantage of a DT in comparison with phrase structure tree (PST) is that the former allows for relating two words directly (and in more compact substructures than PST) even if they are far apart in the corresponding sentence according to their lexical word order.

Several kernel approaches exploit syntactic dependencies among words for PPI extraction from biomedical text in the form of dependency graphs or dependency paths (e.g. Kim et al. (2010) or Airola et al. (2008)). However, to the best of our knowledge, there are only few works on the use of DT kernels for this task. Therefore, exploring the potential of DTKs applied to different structures is a worthwhile research direction. A DTK, pioneered by Culotta and Sorensen (2004), is typically applied to the minimal or smallest common subtree that includes a target pair of entities. Such subtree reduces

Proceedings of the 2011 Workshop on Biomedical Natural Language Processing, ACL-HLT 2011, pages 124–133,
Portland, Oregon, USA, June 23-24, 2011.

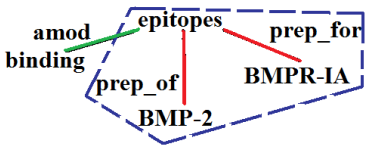

Figure 1: Part of the DT for the sentence "The binding epitopes of *BMP-2* for *BMPR-IA* was characterized using BMP-2 mutant proteins". The dotted area indicates the minimal subtree.

unnecessary information by placing word(s) closer to its dependent(s) inside the tree and emphasizes local features of relations. Nevertheless, there are cases where a minimal subtree might not contain important cue words or predicates. For example, consider the following sentence where a PPI relation holds between *BMP-2* and *BMPR-IA*, but the minimal subtree does not contain the cue word "binding" as shown in Figure 1:

> The binding epitopes of **BMP-2** for **BMPR-IA** was characterized using BMP-2 mutant proteins.

In this paper we investigate two assumptions. The first is that a DTK based on a mild extension of minimal subtrees would produce better results than the DTK on minimal subtrees. The second is that previously proposed DT structures can be further improved by introducing simplified representation of the entities as well as augmenting nodes in the DT tree structure with relevant features. This paper presents an evaluation of the above assumptions.

More specifically, the contributions of this paper are the following:

- We propose the use of new DT structures, which are improvement on the structures defined in Nguyen et al. (2009) with the most general (in terms of substructures) DTK, i.e. Partial Tree Kernel (PTK) (Moschitti, 2006).
- We firstly propose the use of the Unlexicalized PTK (Severyn and Moschitti, 2010) with our dependency structures, which significantly improves PTK.
- We compare the performance of the proposed DTKs on PPI with the one of PST kernels and show that, on biomedical text, DT kernels perform better.
- Finally, we introduce a novel approach (called mildly extended dependency tree (MEDT) kernel[1], which achieves the best performance among various (both DT and PST) tree kernels.

The remainder of the paper is organized as follows. In Section 2, we introduce tree kernels and relation extraction and we also review previous work. Section 3 describes the unlexicalized PTK (uPTK). Then, in Section 4, we define our proposed DT structures including MEDT. Section 5 describes the experimental results on the AIMed corpus (Bunescu et al., 2005) and discusses their outcomes. Finally, we conclude with a summary of our study as well as plans for future work.

2 Background and Related Work

The main stream work for Relation Extraction uses kernel methods. In particular, as the syntactic structure is very important to derive the relationships between entities in text, several tree kernels have been designed and experimented. In this section, we introduce such kernels, the problem of relation extraction and we also focus on the biomedical domain.

2.1 Tree Kernel types

The objective behind the use of tree kernels is to compute the similarity between two instances through counting similarities of their sub-structures. Among the different proposed methods, two of the most effective approaches are Subset Tree (SST) kernel (Collins and Duffy, 2001) and Partial Tree Kernel (PTK) (Moschitti, 2006).

The SST kernel generalizes the subtree kernel (Vishwanathan and Smola, 2002), which considers all common subtrees in the tree representation of two compared sentences. In other words, two subtrees are identical if the node labels and order of children are identical for all nodes. The SST kernel relaxes the constraint that requires leaves to be always included in the sub-structures. In SST, for a given node, either none or all of its children have to be included in the resulting subset tree. An extension of

[1]We defined new structures, which as it is well known it corresponds to define a new kernel.

the SST kernel is the SST+bow (bag-of-words) kernel (Zhang and Lee, 2003; Moschitti, 2006a), which considers individual leaves as sub-structures as well.

The PT kernel (Moschitti, 2006) is more flexible than SST by virtually allowing any tree substructure; the only constraint is that the order of child nodes must be identical. Both SST and PT kernels are convolution tree kernels[2].

The PT kernel is the most complete in terms of structures. However, the massive presence of child node subsequences and single child nodes, which in a DT often correspond to words, may cause overfitting. Thus we propose the use of the unlexicalized (i.e. PT kernel without leaves) tree kernel (uPTK) (Severyn and Moschitti, 2010), in which structures composed by only one lexical element, i.e. single nodes, are removed from the feature space (see Section 3).

2.2 Relation Extraction using Tree Kernels

A first version of dependency tree kernels (DTKs) was proposed by Culotta and Sorensen (2004). In their approach, they find the smallest common subtree in the DT that includes a given pair of entities. Then, each node of the subtree is represented as a feature vector. Finally, these vectors are used to compute similarity. However, the tree kernel they defined is not a convolution kernel, and hence it generates a much lower number of sub-structures resulting in lower performance.

For any two entities *e1* and *e2* in a DT, Nguyen et al. (2009) defined the following three dependency structures to be exploited by convolution tree kernels:

- Dependency Words (DW) tree: a DW tree is the minimal subtree of a DT, which includes *e1* and *e2*. An extra node is inserted as parent of the corresponding NE, labeled with the NE category. Only words are considered in this tree.

- Grammatical Relation (GR) tree: a GR tree is similar to a DW tree except that words are replaced by their grammatical functions, e.g. `prep`, `nsubj`, etc.

- Grammatical Relation and Words (GRW) tree: a GRW tree is the minimal subtree that uses both words and grammatical functions, where the latter are inserted as parent nodes of the former.

Using PTK for the above dependency tree structures, the authors achieved an F-measure of 56.3 (for DW), 60.2 (for GR) and 58.5 (for GRW) on the ACE 2004 corpus[3].

Moschitti (2004) proposed the so called path-enclosed tree (PET)[4] of a PST for Semantic Role Labeling. This was later adapted by Zhang et al. (2005) for relation extraction. A PET is the smallest common subtree of a PST, which includes the two entities involved in a relation.

Zhou et al. (2007) proposed the so called context-sensitive tree kernel approach based on PST, which expands PET to include necessary contextual information. The expansion is carried out by some heuristics tuned on the target RE task.

Nguyen et al. (2009) improved the PET representation by inserting extra nodes for denoting the NE category of the entities inside the subtree. They also used sequence kernels from tree paths, which provided higher accuracy.

2.3 Relation Extraction in the biomedical domain

There are several benchmarks for the PPI task, which adopt different PPI annotations. Consequently the experimental results obtained by different approaches are often difficult to compare. Pyysalo et al. (2008) put together these corpora (including the AIMed corpus used in this paper) in a common format for comparative evaluation. Each of these corpora is known as *converted corpus* of the corresponding original corpus.

Several kernel-based RE approaches have been reported to date for the PPI task. These are based on various methods such as subsequence kernel (Lodhi et al., 2002; Bunescu and Mooney, 2006), dependency graph kernel (Bunescu and Mooney, 2005), etc. Different work exploited dependency analyses with different kernel approaches such as bag-of-

[2]Convolution kernels aim to capture structural information in term of sub-structures, providing a viable alternative to flat features (Moschitti, 2004).

[3]http://projects.ldc.upenn.edu/ace/

[4]Also known as shortest path-enclosed tree or SPT (Zhou et al., 2007).

words kernel (e.g. Miwa et al. (2009)), graph based kernel (e.g. Kim et al. (2010)), etc. However, there are only few researches that attempted the exploitation of tree kernels on dependency tree structures.

Sætre et al. (2007) used DT kernels on AIMed corpus and achieved an F-score of 37.1. The results were far better when they combined the output of the dependency parser with that of a Head-driven Phrase Structure Grammar (HPSG) parser, and applied tree kernel on it. Miwa et al. (2009) also proposed a hybrid kernel [5], which is a composition of all-dependency-paths kernel (Airola et al., 2008), bag-of-words kernel and SST kernel. They used multiple parser inputs. Their system is the current state-of-the-art for PPI extraction on several benchmarks. Interestingly, they applied SST kernel on the shortest dependency paths between pairs of proteins and achieved a relatively high F-score of 55.1. However, the trees they constructed from the shortest dependency paths are actually not dependency trees. In a dependency tree, there is only one node for each individual word whereas in their constructed trees (please refer to Fig. 6 of Miwa et al. (2009)), a word (that belongs to the shortest path) has as many node representations as the number of dependency relations with other words (those belonging to the shortest path). Perhaps, this redundancy of information might be the reason their approach achieved higher result. In addition to work on PPI pair extraction, there has been some approaches that exploited dependency parse analyses along with kernel methods for *identifying sentences* that might contain PPI pairs (e.g. Erkan et al. (2007)).

In this paper, we focus on finding the best representation based on a single structure. We speculate that this can be helpful to improve the state-of-the-art using several combinations of structures and features. As a first step, we decided to use uPTK, which is more robust to overfitting as the description in the next section unveil.

[5] The term "hybrid kernel" is identical to "combined kernel". It refers to those kernels that combine multiple types of kernels (e.g., tree kernels, graph kernels, etc)

3 Unlexicalized Partial Tree Kernel (uPTK)

The uPTK was firstly proposed in (Severyn and Moschitti, 2010) and experimented with semantic role labeling (SRL). The results showed no improvement for such task but it is well known that in SRL lexical information is essential (so in that case it could have been inappropriate). The uPTK definition follows the general setting of tree kernels.

A tree kernel function over two trees, T_1 and T_2, is defined as

$$TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2),$$

where N_{T_1} and N_{T_2} are the sets of nodes in T_1 and T_2, respectively, and

$$\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \chi_i(n_1)\chi_i(n_2).$$

The Δ function is equal to the number of common fragments rooted in nodes n_1 and n_2 and thus depends on the fragment type.

The algorithm for the uPTK computation straightforwardly follows from the definition of the Δ function of PTK provided in (Moschitti, 2006). Given two nodes n_1 and n_2 in the corresponding two trees T_1 and T_2, Δ is evaluated as follows:

1. if the node labels of n_1 and n_2 are different then $\Delta(n_1, n_2) = 0$;
2. else $\Delta(n_1, n_2) = \mu\Big(\lambda^2 + \sum_{\vec{I}_1, \vec{I}_2, l(\vec{I}_1)=l(\vec{I}_2)} \lambda^{d(\vec{I}_1)+d(\vec{I}_2)} \prod_{j=1}^{l(\vec{I}_1)} \Delta(c_{n_1}(\vec{I}_{1j}), c_{n_2}(\vec{I}_{2j}))\Big)$,

where:

1. $\vec{I}_1 = \langle h_1, h_2, h_3, .. \rangle$ and $\vec{I}_2 = \langle k_1, k_2, k_3, .. \rangle$ are index sequences associated with the ordered child sequences c_{n_1} of n_1 and c_{n_2} of n_2, respectively;
2. $\vec{I}_{1j}$ and $\vec{I}_{2j}$ point to the j-th child in the corresponding sequence;
3. $l(\cdot)$ returns the sequence length, i.e. the number of children;

4. $d(\vec{I}_1) = \vec{I}_{1l(\vec{I}_1)} - \vec{I}_{11} + 1$ and $d(\vec{I}_2) = \vec{I}_{2l(\vec{I}_2)} - \vec{I}_{21} + 1$; and

5. μ and λ are two decay factors for the size of the tree and for the length of the child subsequences with respect to the original sequence, i.e. we account for gaps.

The uPTK can be obtained by removing λ^2 from the equation in step 2. An efficient algorithm for the computation of PTK is given in (Moschitti, 2006). This evaluates Δ by summing the contribution of tree structures coming from different types of sequences, e.g. those composed by p children such as:

$$\Delta(n_1, n_2) = \mu\big(\lambda^2 + \sum_{p=1}^{lm} \Delta_p(c_{n_1}, c_{n_2})\big), \quad (1)$$

where Δ_p evaluates the number of common subtrees rooted in subsequences of exactly p children (of n_1 and n_2) and $lm = min\{l(c_{n1}), l(c_{n2})\}$. It is easy to verify that we can use the recursive computation of Δ_p by simply removing λ^2 from Eq. 1.

4 Proposed dependency structures and MEDT kernel

Our objective is twofold: (a) the definition of improved DT structures and (b) the design of new DT kernels to include important words residing outside of the shortest dependency tree, which are neglected in current approaches. For achieving point (a), we modify the DW, GR and GRW structures, previously proposed by Nguyen et al. (2009). The new proposed structures are the following:

- Grammatical Relation and lemma (GRL) tree: A GRL tree is similar to a GRW tree except that words are replaced by their corresponding lemmas.

- Grammatical Relation, PoS and lemma (GRPL) tree: A GRPL tree is an extension of a GRL tree, where the part-of-speech (PoS) tag of each of the corresponding words is inserted as a new node between its grammatical function and its lemma, i.e. the new node becomes the parent node of the node containing the lemma.

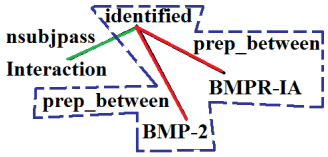

Figure 2: Part of the DT for the sentence "Interaction was identified between *BMP-2* and *BMPR-IA*". The dotted area indicates the minimal subtree.

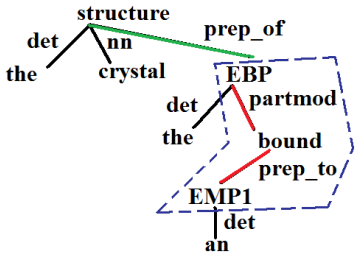

Figure 3: Part of the DT for the sentence "Phe93 forms extensive contacts with a peptide ligand in the crystal structure of the *EBP* bound to an *EMP1*". The dotted area indicates the minimal subtree.

- Ordered GRL (OGRL) or ordered GRW (OGRW) tree: in a GRW (or GRL) tree, the node containing the grammatical function of a word is inserted as the parent node of such word. So, if the word has a parent node containing its NE category, the newly inserted node with grammatical function becomes the child node of the node containing NE category, i.e. the order of the nodes is the following – *"NE category ⇒ grammatical relation ⇒ word (or lemma)"*. However, in OGRW (or OGRL), this ordering is modified as follows – *"grammatical relation ⇒ NE category ⇒ word (or lemma)"*.

- Ordered GRPL (OGRPL) tree: this is similar to the OGRL tree except for the order of the nodes, which is the following – *"grammatical relation ⇒ NE category ⇒ PoS ⇒ lemma"*.

- Simplified (S) tree: any tree structure would become an S tree if it contains simplified representations of the entity types, where all its parts except the head word of a multi-word entity are not considered in the minimal subtree.

The second objective is to extend DTKs to include important cue words or predicates that are missing

in the minimal subtree. We do so by mildly expanding the minimal subtree, i.e. we define the mildly extended DT (MEDT) kernel. We propose three different expansion rules for three versions of MEDT as follows:

- Expansion rule for MEDT-1 kernel: *If the root of the minimal subtree is not a modifier (e.g. adjective) or a verb, then look for such node in its children or in its parent (in the original DT tree) to extend the subtree.*

  The following example shows a sentence where this rule would be applicable:

  > The binding epitopes of **BMP-2** for **BMPR-IA** was characterized using BMP-2 mutant proteins.

  Here, the cue word is "binding", the root of the minimal subtree is "epitopes" and the target entities are *BMP-2* and *BMPR-IA*. However, as shown in Figure 1, the minimal subtree does not contain the cue word.

- Expansion rule for MEDT-2 kernel: *If the root of the minimal subtree is a verb and its subject (or passive subject) in the original DT tree is not included in the subtree, then include it.*

  Consider the following sentence:

  > Interaction was identified between **BMP-2** and **BMPR-IA**.

  Here, the cue word is "Interaction", the root is "identified" and the entities are *BMP-2* and *BMPR-IA*. The passive subject "Interaction" does not belong to the minimal subtree (see Figure 2).

- Expansion rule for MEDT-3 kernel: *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.*

  This is an example sentence where this rule is applicable (see Figure 3):

  > Phe93 forms extensive contacts with a peptide ligand in the crystal structure of the **EBP** bound to an **EMP1**.

5 Experiments and results

We carried out several experiments with different dependency structures and tree kernels. Most importantly, we tested tree kernels on PST and our improved representations for DT.

5.1 Data and experimental setup

We used the AIMed corpus (Bunescu et al., 2005) converted using the software provided by Pyysalo et al. (2008). AIMed is the largest benchmark corpus (in terms of number of sentences) for the PPI task. It contains 1,955 sentences, in which are annotated 1,000 positive PPI and 4,834 negative pairs.

We use the Stanford parser[6] for parsing the data.[7] The SPECIALIST lexicon tool[8] is used to normalize words to avoid spelling variations and also to provide lemmas. For training and evaluating tree kernels, we use the SVM-LIGHT-TK toolkit[9] (Moschitti, 2006; Joachims, 1999). We tuned the parameters μ, λ and c following the approach described by Hsu et al. (2003), and used biased hyperplane.[10] All the other parameters are left as their default values.

Our experiments are evaluated with 10-fold cross validation using the same split of the AIMed corpus used by Bunescu et al. (2005).

5.2 Results and Discussion

The results of different tree kernels applied to different structures are shown in Tables 1 and 2. All the tree structures are tested with four different tree kernel types: SST, SST+bow, PTK and uPTK.

According to the empirical outcome, our new DT structures perform better than the existing tree structures. The highest result (F: 46.26) is obtained by applying uPTK to MEDT-3 (SOGRL). This is 6.68 higher than the best F-measure obtained by previous DT structures proposed in Nguyen et al. (2009), and 0.36 higher than the best F-measure obtained using PST (PET).

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

[7] For some of the positive PPI pairs, the connecting dependency tree could not be constructed due to parsing errors for the corresponding sentences. Such pairs are considered as false negative (FN) during precision and recall measurements.

[8] http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html

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

[10] Please refer to http://svmlight.joachims.org/ and http://disi.unitn.it/moschitti/Tree-Kernel.htm for details about parameters of the respective tools

	DT (GR)	DT (SGR)	DT (DW)	DT (SDW)	DT (GRW)	DT (SGRW)	DT (SGRL)	DT (SGRPL)	DT (OGRPL)
SST	P: 55.29	P: 54.22	P: 31.87	P: 30.74	P: 52.76	P: 52.47	P: 56.09	P: 56.03	P: 57.85
	R: 23.5	R: 24.4	R: 27.5	R: 27.3	R: 33.4	R: 30.8	R: 33.6	R: 33.0	R: 31.7
	F: 32.98	F: 33.66	F: 29.52	F: 28.92	F: 40.9	F: 38.82	F: 42.03	F: 41.54	F: 40.96
SST	P: 57.87	P: 54.91	P: 30.71	P: 29.98	P: 52.98	P: 51.06	P: 51.99	P: 56.8	P: 61.73
+	R: 21.7	R: 23.5	R: 26.9	R: 25.9	R: 32.0	R: 31.3	R: 31.4	R: 28.8	R: 29.2
bow	F: 31.56	F: 32.91	F: 28.68	F: 27.79	F: 39.9	F: 38.81	F: 39.15	F: 38.22	F: 39.65
PT	P: 60.0	P: 57.84	P: 40.44	P: 42.2	P: 53.35	P: 53.41	P: 51.29	P: 52.88	P: 53.55
	R: 15.9	R: 16.6	R: 23.9	R: 26.5	R: 34.2	R: 36.0	R: 37.9	R: 33.0	R: 33.2
	F: 25.14	F: 25.8	F: 30.04	F: 32.56	F: 41.68	F: 43.01	F: 43.59	F: 40.64	F: 40.99
uPT	P: 58.77	P: 59.5	P: 29.21	P: 29.52	P: 51.86	P: 52.17	P: 52.1	P: 54.64	P: 56.43
	R: 23.8	R: 26.0	R: 30.2	R: 31.5	R: 32.0	R: 33.7	R: 36.0	R: 31.2	R: 30.7
	F: 33.88	F: 36.19	F: 29.7	F: 30.48	F: 39.58	F: 40.95	F: 42.58	F: 39.72	F: 39.77

Table 1: Performance of DT (GR), DT (DW) and DT (GRW) (proposed by (Nguyen et al., 2009)) and their modified and improved versions on the *converted* AIMed corpus.

RE experiments carried out on newspaper text corpora (such as ACE 2004) have indicated that kernels based on PST obtain better results than kernels based on DT. Interestingly, our experiments on a biomedical text corpus indicate an opposite trend. Intuitively, this might be due to the different nature of the PPI task. PPI can be often identified by spotting cue words such as *interaction*, *binding*, etc, since the interacting entities (i.e. proteins) usually have direct syntactic dependency relation on such cue words. This might have allowed kernels based on DT to be more accurate.

Although tree kernels applied on DT and PST structures have produced high performance on corpora of news text (Zhou et al., 2007; Nguyen et al., 2009), in case of biomedical text the results that we obtained are relatively low. This may be due to the fact that biomedical texts are different from newspaper texts: more variation in vocabulary, more complex naming of (bio) entities, more diversity of the valency of verbs and so on.

One important finding of our experiments is the effectiveness of the mild extension of DT structures. MEDT-3 achieves the best result for all kernels (SST, SST+bow, PTK and uPTK). However, the other two versions of MEDT appear to be less effective.

In general, the empirical outcome suggests that uPTK can better exploit our proposed DT structures as well as PST. The superiority of uPTK on PTK demonstrates that single lexical features (i.e. features with flat structure) tend to overfit.

Finally, we have performed statistical tests to assess the significance of our results. For each kernel (i.e. SST, SST+bow, PTK, uPTK), the PPI predictions using the best structure (i.e. MEDT-3 applied to SOGRL) are compared against the predictions of the other structures. The tests were performed using the approximate randomization procedure (Noreen, 1989). We set the number of iterations to 1,000 and the confidence level to 0.01. According to the tests, for each kernel, our best structure produces significantly better results.

5.3 Comparison with previous work

To the best of our knowledge, the only work on tree kernel applied on dependency trees that we can directly compare to ours is reported by Sætre et al. (2007). Their DT kernel achieved an F-score of 37.1 on AIMed corpus which is lower than our best results. As discussed earlier, Miwa et al. (2009)) also used tree kernel on dependency analyses and achieved a much higher result. However, the tree structure they used contains multiple nodes for a single word and this does not comply with the constraints usually applied to dependency tree structures (refer to Section 2.3). It would be interesting to examine why such type of tree representation leads to

	DT (SOGRPL)	DT (OGRL)	DT (SOGRW)	DT (SOGRL)	MEDT-1 (SOGRL)	MEDT-2 (SOGRL)	MEDT-3 (SOGRPL)	PST (PET)
SST	P: 57.59	P: 54.38	P: 51.49	P: 54.08	P: 58.15	P: 54.46	P: 59.55	P: 52.72
	R: 33.0	R: 33.5	R: 31.2	R: 33.8	R: 34.6	R: 33.6	R: 37.1	R: 35.9
	F: 41.96	F: 41.46	F: 38.86	F: 41.6	F: 43.39	F: 41.56	F: 45.72	F: 42.71
SST	P: 60.31	P: 53.22	P: 50.08	P: 53.26	P: 58.84	P: 52.87	P: 59.35	P: 52.88
+	R: 30.7	R: 33.1	R: 30.9	R: 32.7	R: 32.6	R: 32.2	R: 34.9	R: 37.7
bow	F: 40.69	F: 40.82	F: 38.22	F: 40.52	F: 41.96	F: 40.02	F: 43.95	F: 44.02
PT	P: 55.45	P: 49.78	P: 51.05	P: 51.61	P: 52.94	P: 50.89	P: 54.1	P: 58.39
	R: 34.6	R: 34.6	R: 34.1	R: 36.9	R: 36.0	R: 37.0	R: 38.9	R: 36.9
	F: 42.61	F: 40.82	F: 40.89	F: 43.03	F: 42.86	F: 42.85	F: 45.26	F: 45.22
uPT	P: 56.2	P: 50.87	P: 50.0	P: 52.74	P: 55.0	P: 52.17	P: 56.85	P: 56.6
	R: 32.2	R: 35.0	R: 33.0	R: 35.6	R: 34.1	R: 34.8	R: 39.0	R: 38.6
	F: 40.94	F: 41.47	F: 39.76	F: 42.51	F: 42.1	F: 41.75	**F: 46.26**	F: 45.9

Table 2: Performance of the other improved versions of DT kernel structures (including MEDT kernels) as well as PST (PET) kernel (Moschitti, 2004; Nguyen et al., 2009) on the *converted* AIMed corpus.

a better result.

In this work, we compare the performance of tree kernels applied of DT with that of PST. Previously, Tikk et al. (2010) applied similar kernels on PST for exactly the same task and data set. They reported that SST and PTK (on PST) achieved F-scores of 26.2 and 34.6, respectively on the converted AIMed corpus (refer to Table 2 in their paper). Such results do not match our figures obtained with the same kernels on PST. We obtain much higher results for those kernels. It is difficult to understand the reason for such differences between our and their results. A possible explanation could be related to parameter settings. Another source of uncertainty is given by the tool for tree kernel computation, which in their case is not mentioned. Moreover, their description of PT and SST (in Figure 1 of their paper) appears to be imprecise: for example, in (partial or complete) phrase structure trees, words can only appear as leaves but in their figure they appear as non-terminal nodes.

The comparison with other kernel approaches (i.e. not necessarily tree kernels on DT or PST) shows that there are model achieving higher results (e.g. Giuliano et al. (2006), Kim et al. (2010), Airola et al. (2008), etc). State-of-the-art results on most of the PPI data sets are obtained by the hybrid kernel presented in Miwa et al. (2009). As noted earlier, our work focuses on the design of an effective DTK for PPI that can be combined with others and that can hopefully be used to design state-of-the-art hybrid kernels.

6 Conclusion

In this paper, we have proposed a study of PPI extraction from specific biomedical data based on tree kernels. We have modeled and experimented with new kernels and DT structures, which can be exploited for RE tasks in other domains too.

More specifically, we applied four different tree kernels on existing and newly proposed DT and PST structures. We have introduced some extensions of DT kernel structures which are linguistically motivated. We call these as mildly extended DT kernels. We have also shown that in PPI extraction lexical information can lead to overfitting as uPTK outperforms PTK. In general, the empirical results show that our DT structures perform better than the previously proposed PST and DT structures.

The ultimate objective of our work is to improve tree kernels applied to DT and then combine them with other types of kernels and data to produce more accurate models.

Acknowledgments

This work was carried out in the context of the project "eOnco - Pervasive knowledge and data management in cancer care". The authors would like to thank the anonymous reviewers for providing excellent feedback.

References

A Airola, S Pyysalo, J Bj̈orne, T Pahikkala, F Ginter, and T Salakoski. 2008. A graph kernel for protein-protein interaction extraction. In *Proceedings of BioNLP 2008*, pages 1–9, Columbus, USA.

R Bunescu and R Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

R Bunescu and RJ Mooney. 2006. Subsequence kernels for relation extraction. In *Proceedings of the 19th Conference on Neural Information Processing Systems*, pages 171–178.

R Bunescu, R Ge, RJ Kate, EM Marcotte, RJ Mooney, AK Ramani, and YW Wong. 2005. 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.

M Collins and N Duffy. 2001. Convolution kernels for natural language. In *Proceedings of Neural Information Processing Systems (NIPS'2001)*.

A Culotta and J Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain.

G Erkan, A Ozgur, and DR Radev. 2007. Semi-Supervised Classification for Extracting Protein Interaction Sentences using Dependency Parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 228–237.

C Giuliano, A Lavelli, and L Romano. 2006. 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)*, pages 401–408, Trento, Italy.

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.

T 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.

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

H Lodhi, C Saunders, J Shawe-Taylor, N Cristianini, and C Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, March.

M Miwa, R Sætre, Y Miyao, T Ohta, and J Tsujii. 2009. Protein-protein interaction extraction by leveraging multiple kernels and parsers. *International Journal of Medical Informatics*, 78.

A Moschitti. 2004. 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.

A Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, volume 4212 of *Lecture Notes in Computer Science*, pages 318–329. Springer Berlin / Heidelberg.

A Moschitti. 2006a. Making Tree Kernels Practical for Natural Language Learning. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.

TT Nguyen, A Moschitti, and G Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP'2009)*, pages 1378–1387, Singapore, August.

EW Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience.

S Pyysalo, A Airola, J Heimonen, J Björne, F Ginter, and T Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3):S6.

R Sætre, K Sagae, and J Tsujii. 2007. Syntactic features for protein-protein interaction extraction. In *Proceedings of the Second International Symposium on Languages in Biology and Medicine (LBM 2007)*, pages 6.1–6.14, Singapore.

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

D Tikk, P Thomas, P Palaga, J Hakenberg, and U Leser. 2010. A Comprehensive Benchmark of Kernel Methods to Extract Protein-Protein Interactions from Literature. *PLoS Computational Biology*, 6(7), July.

SVN Vishwanathan and AJ Smola. 2002. Fast kernels on strings and trees. In *Proceedings of Neural Information Processing Systems (NIPS'2002)*, pages 569–576, Vancouver, British Columbia, Canada.

D Zhang and WS Lee. 2003. Question classification using support vector machines. In *Proceedings of the*

26th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '03, pages 26–32, Toronto, Canada.

M Zhang, J Su, D Wang, G Zhou, and CL Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, editors, *Natural Language Processing IJCNLP 2005*, volume 3651 of *Lecture Notes in Computer Science*, pages 378–389. Springer Berlin / Heidelberg.

GD Zhou, M Zhang, DH Ji, and QM Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 728–736, June.