An extended dependency graph for relation extraction in biomedical texts

Yifan Peng[1] **Samir Gupta**[1] **Cathy H. Wu**[1,2] **K. Vijay-Shanker**[1]
[1]Department of Computer Information & Sciences
[2]Center for Bioinformatics and Computational Biology
University of Delaware
Newark, DE 19716
{yfpeng,sgupta,wuc,vijay}@udel.edu

Abstract

Kernel-based methods are widely used for relation extraction task and obtain good results by leveraging lexical and syntactic information. However, in biomedical domain these methods are limited by the size of dataset and have difficulty in coping with variations in text. To address this problem, we propose Extended Dependency Graph (EDG) by incorporating a few simple linguistic ideas and include information beyond syntax. We believe the use of EDG will enable machine learning methods to generalize more easily. Experiments confirm that EDG provides up to 10% f-value improvement over dependency graph using mainstream kernel methods over five corpora. We conducted additional experiments to provide a more detailed analysis of the contributions of individual modules in EDG construction.

1 Introduction

With growing amount of biomedical information available in textual form, there has been considerable interest in applying NLP techniques and machine-learning (ML) methods to biomedical literature. Some of these projects involve extracting relations such as protein-protein interaction (Krallinger et al., 2008).

In biomedical domain, most relation extraction work is currently applied on the abstracts of articles. These abstracts by nature are dense with information and often use constructions such as appositives and relative clauses. The abundance of textual variations can thus be problematic for ML systems, especially with small training corpora.

One solution to this issue is to find a suitable level of abstraction in the text representation so that ML methods become easier to generalize. Use of syntax and parse information provides one such abstraction. Using syntactic dependency information has become prevalent in biomedical relation extraction. It has been suggested dependency links are close to the semantic relationship needed for the next stage of interpretation (Covington, 2001).

There have been significant advances in the development of advanced machine learning and kernel methods and the use of sophisticated parameter tuning in the biomedical domain. In this work, we focus on the representation of the text used in learning rather than the machine learning technique, with the hope that advances in both directions will be improve the performance of the relation extraction systems. In this paper we propose Extended Dependency Graph (EDG), which includes information about text that goes beyond syntax. We will define EDG and discuss how we construct it from a given sentence by using some simple linguistic notions.

The hypothesis we test here is that EDG allows ML techniques to generalize more easily. To determine the effect of EDG, we conducted experiments on protein-protein interaction (PPI) extraction. For this purpose, we used two kernels: a simple kernel based on edit distance (Erkan et al., 2007) and a more elaborate kernel that is one of the top performing kernels on the PPI task (Airola et al., 2008). We compared the performance of both kernels using dependency graph and EDG on 5 corpora. Our results suggest EDG provides up to 10% f-value improvement over dependency graph. On 3 out of 5 corpora the results are better than the overall best system in the study of (Tikk et al., 2010), as well as an ensemble method that builds on them (Miwa et al., 2009a). We also evaluate the contributions of the individual components included in EDG.

Proceedings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP 2015), pages 21–30,
Beijing, China, July 30, 2015.

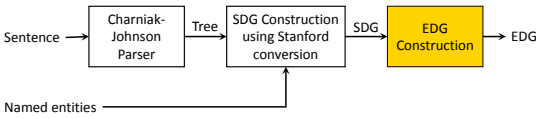

Figure 1: Framework.

2 Related work

Many kernel-based relation extraction systems have employed lexical and syntactic information (Bunescu and Mooney, 2005; Zhou et al., 2007; Ning and Qi, 2011). There has been a growth in the use of more complex kernels and sophisticated parameter tuning methods to improve the results (Zhang et al., 2006; Choi and Myaeng, 2010). In PPI task, machine learning methods using rich feature vectors (Miwa et al., 2009b), edit distance kernel (Erkan et al., 2007), dependency tree kernel (Chowdhury et al., 2011), all-path graph kernel (Airola et al., 2008), or their combination and variations (Miwa et al., 2009a; Zhang et al., 2012) have been proposed.

Our focus is on improving the representation of information in natural texts, rather than on developing new kernels. There have been several attempts to leverage syntax and shallow semantic argument structure (Miwa et al., 2010; Van Landeghem et al., 2010; Van Landeghem et al., 2012; Liu et al., 2013; Oepen et al., 2014; Peng et al., 2014; Nguyen et al., 2015). Though the focus of these works was not to utilize the information with machine learning methods, they offer insight on utility of information beyond syntax. We develop the EDG approach for relation extraction based on these ideas.

3 Method

Figure 1 illustrates the overall architecture with the core component highlighted: EDG construction. The input is a sentence with named entities marked. We use Charniak-Johnson parser and Stanford conversion tool to get the basic syntactic dependency graph (SDG). Our approach focuses on how to leverage simple linguistic principles and information beyond syntax to construct EDG from SDG.

3.1 Extended dependency graph (EDG)

In this paper, we use EDG to represent the structure of the sentence. Like in the case of many dependency graph representations used in relation extraction, the vertices in a EDG are labelled with information such as the text, part-of-speech, and the word lemma. If an entity mention spans multiple tokens in a sentence, we merge their corresponding vertices (called contracting vertices) into one vertex.

EDG has two types of dependencies. The syntactic dependencies that are obtained from collapsed dependencies output by applying Stanford dependencies converter on a syntactic parsing tree (De Marneffe and Manning, 2008). The other type of dependencies are the numbered arguments based on the guidelines of PropBank (Bonial et al., 2012). Because we are currently focusing on binary relation extraction, we use only *arg0* and *arg1* (probably better stated as not-*arg0*) in EDG. Figure 2 shows EDGs of three text fragments with syntactic edges appearing above the words and numbered argument edges appearing below. From a relation extraction perspective, the syntactic dependencies in Figure 2 are less relevant but their numbered arguments between two entity mentions are same.

There are two motivations for using numbered arguments. One is to "provide consistent argument labels across different syntactic realizations of the same verb" (Bonial et al., 2012) with the intention of making generalizations easier downstream. The other is to add/propagate new *arg0* and *arg1* using reasoning that goes beyond syntax.

Following these two motivations, we will first discuss how to capture *arg0* and *arg1* using different syntactic dependencies obtained from Stanford dependencies. Then we will describe relations such as *is-a*, *member-collection*, and *part-whole* and how to propagate *arg0* and *arg1* using them.

3.2 Syntax based *arg0* and *arg1*

We follow approaches of SemRep (Rinaldi et al., 2006) and PASMED (Nguyen et al., 2015) to obtain the basic edges *arg0* and *arg1* from the syntactic dependencies. For example, EDG will include an *arg0* from a verb to the noun if the syntactic dependency is *nsubj* or *agent* and include an *arg1* if the dependency is *nsubjpass* or *dobj*.

In addition, we consider situation where verbs in gerund form are used as noun modifiers. Figure 3 shows a compound noun phrase. We know that there is a PPI between "retinoblastoma" and "protein", because we can rewrite the phrase into

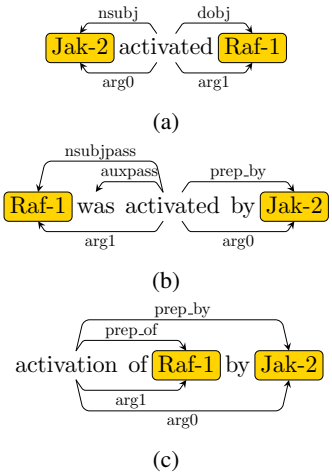

Figure 2: Sample EDGs with an active (a), passive (b), or normalized (c) verb.

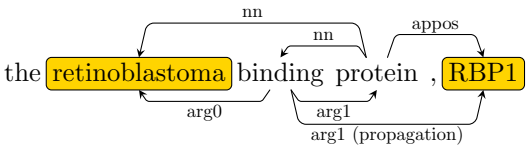

Figure 3: A sample compound noun phrase.

"retinoblastoma binds to protein, RBP1". Therefore, we add *arg1* from "binding" to "protein" in Figure 3. This operation will introduce cyclicity because the gerund is included in the noun phrase headed by "protein". We posit that these edges are useful when found in combination with other construction, such as appositive. We will discuss how to propagate *arg1* from the gerund "binding" to "RBP1" later.

Next we consider two cases of argument elision.

Elided argument relation Here we consider cases when the argument of a predicate is not explicit but implicit. Figure 4 shows a sentence where *arg0(interaction, Presenilin 1)* can be inferred. The SDG includes a *prep_via* from the first verb "suppresses" to the nominalized verb "interaction", to indicate the PP attachment to the verb. In this case, we add an edge *arg0* from the nominalized verb to the *arg0*-argument of the first verb. In constructing EDG, we also consider *prep_through* as well as *prep_by* when a gerund verb, rather than a nominalized verb, follows it.

Reduced relative clauses Relative clause is a clause that modifies a noun phrase. There are two types of relative clauses that frequently appear in biomedical text. Full relative clauses are introduced by relative pronouns, such as "which"

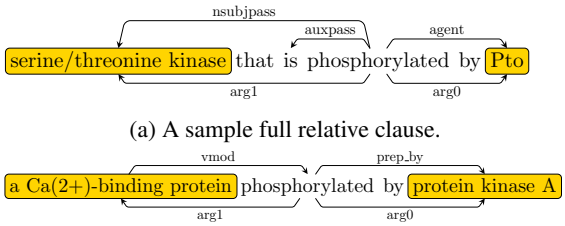

Figure 5: Sample relative clauses.

and "that". Reduced relative clauses start with a gerund or past participle and have no overt subject.

The PropBank annotation guidelines (Bonial et al., 2012) posit a numbered argument link from the relative clause verb to the trace in the parse tree which also indicates the referent noun phrase. For full relative clauses, we follow the normal procedure for verbs (Figure 5a). For reduced relative clauses, since we use the dependency structure that includes no traces, we use the edge *vmod* in the SDG from the head of the noun phrase to the reduced relative clause's verb (Figure 5b). The direction of this edge indicates that the relative clause is syntactically included in the larger noun phrase. For the arg edge, we reverse the direction of *vmod* and create an edge from the relative clause's verb, as shown in Figure 5b. When compared to Figure 5a, the *arg* construction unifies the treatment for full relative clauses.

Notice that although in both cases, the *arg1* is not an incident on named entities, it might still lead to the named entity through the propagation of edges as discussed in the next subsection.

3.3 Going Beyond Syntax

Here we consider the propagation of *arg* using information that goes beyond syntax.

Co-reference If an edge *arg* from a vertex *v* reaches a pronominal node, we add a new edge *arg* from *v* to any named entity the pronoun coreferes to. To detect the coreference we use the implementation of the technique described in (Qiu et al., 2004). For the acronyms with long-form and short-form, we treat them in the same way as coreference. We add extra edge *arg* when there is an *arg* incident on the long-form. We use the acronym detector of (Schwartz and Hearst, 2003) to add acronyms missed in SDG. Interestingly, SDG uses *appos* for both acronym and appositive.

Appositive Reconsider the fragment "the

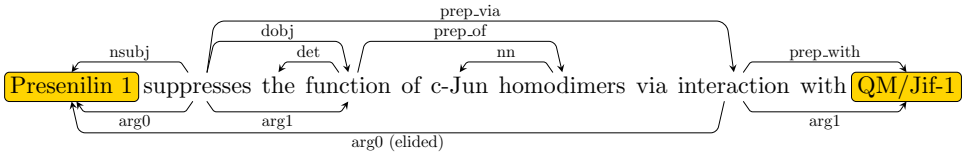

Figure 4: A sample elided argument relation.

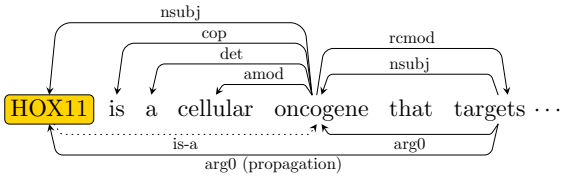

Figure 6: A sample *is-a* relation.

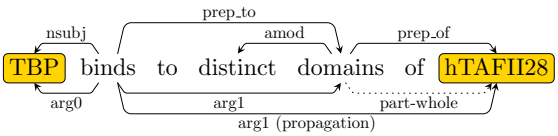

Figure 8: A sample *part-whole* relation.

retinoblastoma binding protein, RBP1" in Figure 3. Using the construction discussed thus far, the *arg1* will reach "protein". Further, SDG uses an edge *appos* from "protein" to "RBP1" for appositional modifier. We integrate *arg1* and *appos* to construct another edge *arg1* from "binding" to the actual named entity "RBP1".

Is-A In addition to appositive, we consider other forms of *is-a* relation mentioned textually, but cannot be directly found from the syntactic dependences. For example, in Figure 6, there is no edge in SDG to explicitly capture the *is-a* relation. It is worth noting that the edge *nsubj* itself does not indicate the *is-a* relation, but together with two other edges *cop* and *det*, we can figure it out. Hence we add a new edge from "oncogene" to "HOX11" to reflect this relation in EDG (dotted edge). Afterwards, we propagate *arg0* from "targets" to "HOX11".

Besides the pattern shown in Figure 6, we also identify "known as", "designated as", "considered as", "identified as" and "act as" as patterns that signal *is-a* relations. These patterns contain and extend rules in (Snow et al., 2005; Hearst, 1992).

Member-collection links a generic reference (called *collection*) to a group of entity mentions (called *members*). Like in Figure 7, typical key words that can identify *member-collection* relations are "including" and "such as". We consider the cases where mention group follows the keywords and the generic reference precedes these words. After the detection, we propagate *arg* from the collection to its members.

Part-whole links an entity part to its mention, typically denoting construction of larger entities out of smaller ones. Just like "breaking the glass of the window" can be stated as "breaking the window", in biomedical tasks an action on a larger unit can often be inferred from a mention of the action applied on its part. That is, in Figure 8, after we detect a *part-whole* relation, an edge *arg1* incident on the part is propagated to the object that contains it.

In this paper, we focus on three types of patterns to recognize *part-whole* relations. The first is the preposition phrase such as "domain of *e*". Here "domain" indicates the part and *e* indicates the larger entity mention the "domain" belongs to. Other keywords indicating parts include "fragment", "portion", and "region". The second structural elements is a compound nominal like "*e* domain". The third group exploits keywords such as "contain", "consist", and "compose". For each *part-whole* relation, we propagate edges from the part to its entity mention.

4 Experiments

We evaluated our method on protein-protein interaction (PPI) extraction task, where the system identifies whether a given protein pair in a sentence has PPI relationship or not. We used SDG or EDG as input representation of the sentences, which includes the named protein entities.

4.1 Kernels

We tested the effect of using EDG on two kernels that have been employed for PPI extraction.

Edit distance kernel is based on the edit distance among the shortest paths between entities in the dependency graph and is based on the minimal number of operations (deletion, insertion, substitution at word level) needed to transform one path (p_1) into the other (p_2). Following (Erkan et al.,

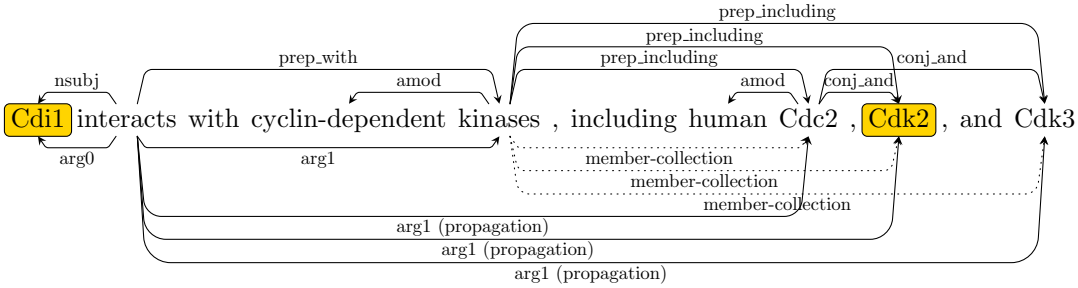

Figure 7: A sample *member-collection* relation.

2007), this number is normalized by the length of the longer path and converted into a similarity measure.

$$sim_e(p_1, p_2) = e^{-\gamma \text{editdist}(p_1, p_2)} \quad (1)$$

When comparing two shortest paths, we considered the word lemma and the edge labels . We also renamed the candidate pair in the sentence as "E1" and "E2" and the remaining proteins provided in the annotation as "EX". For example, the following are the shortest paths of Figure 2a, 3, and 8.

(a) E1 $\leftarrow$ *arg0* $\leftarrow$ activate $\rightarrow$ *arg1* $\rightarrow$ E2
(b) E1 $\leftarrow$ *arg0* $\leftarrow$ bind $\rightarrow$ *arg1* $\rightarrow$ E2
(c) E1 $\leftarrow$ *arg0* $\leftarrow$ bind $\rightarrow$ *arg1* $\rightarrow$ E2

Therefore, the edit distance between (a) and (b) is 1 because the predicate verbs are different. The distance between (b) and (c) is 0. It shows the generalizability of using EDG.

All-paths graph kernel is a practical instantiation of a graph kernel framework (Gärtner et al., 2003). It counts weighted shared paths of all possible lengths in a graph (Airola et al., 2008). All-paths graph kernel uses two graph representations: (1) a dependency graph where all edges on the shortest paths between the candidate pair receive a weight of 0.9 and other edges receive a weight of 0.3; and (2) a linear graph where each word node is connected by an edge to its succeeding word node with weight 0.9.

We used word (not lemma) and edge labels to compute the all-paths graph kernel. Similar to the case with the edit distance kernel, we replaced the protein names in a sentence with "E1", "E2" and "EX". We use the APG software (`http://mars.cs.utu.fi/PPICorpora/GraphKernel.html`) to train and test the kernel. The software uses sparse regularized least squares method instead of SVM.

Table 1: Basic statistics of the corpora.

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

4.2 Experimental setup

We evaluated our method on five PPI corpora that have been used in the community: AIMed (Bunescu et al., 2005), BioInfer (Pyysalo et al., 2007), HPRD50 (Fundel et al., 2007), IEPA (Ding et al., 2002), and LLL (Nédellec, 2005). These corpora have different sizes (Table 1) and vary slightly in their definition of PPI (Pyysalo et al., 2008).

(Tikk et al., 2010) conducted a comparison of a variety of PPI extraction systems on these corpora (`http://mars.cs.utu.fi/PPICorpora`). We used the same experimental setup to evaluate our methods: self-interactions were excluded from the corpora and 10-fold document-level cross-validation is used for evaluation.

For our experiments, we used the Charniak-Johnson parser (Charniak and Johnson, 2005) and the Stanford conversion tool with "Collapsed" setting to obtain SDG (De Marneffe and Manning, 2008). The edit distance kernel was trained with LIBSVM (Chang and Lin, 2011). The APG kernel was trained with APG software.

Both these kernels have several parameters, whose settings can influence the performance. In this paper, we did not perform exhaustive systematic parameter search and optimization. We believe such parameter tuning techniques might lead to further improvements.

For the edit kernel, we set γ to 4.5, which was

the value used in the original application of edit kernel on these corpora (Erkan et al., 2007). We set c in SVM to 10, which was the average best value used in (Tikk et al., 2010). For the APG kernel, we used the default settings of implementation of (Airola et al., 2008) which uses a grid parameter search for each iteration of the 10-fold cross validation. The parameter search selects the best setting based on a random set of 1,000 samples from the training sets (9 folds). If there are less than 1,000 samples, the software used the whole training set. Note that the test sets (the remaining fold) were not used for the parameter tuning.

4.3 Results

Performance, as measured by precision, recall, and F-value, is shown in Table 2. To provide context, we also include the results published in (Tikk et al., 2010) and (Miwa et al., 2009a). The first reports the results of the APG kernel (Airola et al., 2008) that was found to be a leading performer on these 5 corpora in the study reported in (Tikk et al., 2010). The second set of results is those of an ensemble method that combines different systems.

Although we are using the same corpora in the study of (Tikk et al., 2010), and the same implementation of the APG kernel, the results in Row 1 and Row 6 in the table are not the same. The differences are possibly due to the fact that different parsers were used and how parameters were chosen. However, we want to emphasize that all our own measurements (e.g., in Rows 3-5 or Rows 6-8) are directly comparable to each other because the same parameter settings were used for each corpus.

The first part of Table 2 shows results using the edit distance kernel with original dependency graph (Row 3), and with the complete EDG (Row 4). We also experimented with different configurations of EDG by dropping one of the extra edge types added in EDG. The results obtained by the best configuration are reported in Row 5. On three of the corpora, the best results are obtained by using the full EDG. However, better results were obtained on HPRD50, when the *member-collection* relations were not included and on LLL, when the *is-a* relations were not included. In the next subsection we will address why these relations were not included.

Overall, comparing Rows 3 and 4, we obtain F-value improvements using EDG over using SDG on 4 corpora (except LLL), with around 10% gains on AIMed and HPRD50 and noticeable gain in recall. For 3 of the corpora (AIMed, HPRD50 and IEPA), there is an increase in both precision and recall. For BioInfer, the gain in precision slightly exceeds the loss in recall whereas in LLL the gain in precision is slightly lower than the loss in recall. When Row 5 is used for comparison, we obtain an improvement in F-value for all 5 corpora with improvement in recision and recall in 4 corpora (BioInfer being the exception). We now see over 18% F-value improvement on HPRD50.

Despite weak performance of the edit kernel using the baseline SDG, the performance of this kernel with full EDG is close to or exceeds the results of the leading PPI systems using kernel methods (Rows 1 and 2) on 4 corpora and exceeds them on these 4 corpora when results of Row 5 is considered.

The second part of Table 2 (Rows 6–8) shows results using the APG kernel. The EDG (Best) in Row 8 is achieved on AIMed, BioInfer and LLL by dropping the *is-a* relation and on HPRD50 by not including the *member-collection* relations. We see F-value gains on 4 corpora through the use of EDG.

Comparing the results on the edit distance and APG kernels, we find that the more complex APG kernel (the best one overall in (Tikk et al., 2010) study) gets generally better results than Edit kernel using the baseline SDG. However, the use of EDG not only closes the gap between the kernels but in fact, edit kernel with EDG obtains higher F-value than APG with SDG or EDG in 4 of the 5 corpora.

To provide the comparision with non-kernel methods, we also include the results published in (Miwa et al., 2009b), which is the state-of-the-art system on the five corpora. This paper develops several systems that use a rich feature vector, combining analysis from different parsers and the values obtained from multiple kernels including the APG's score. L2-SVM and SVM-CW are among the leading SVM-based systems proposed in this paper.

Row 9 shows the results of L2-SVM on these corpora. We observe that both edit kernel and APG kernel with EDG (Best) gets improvements on two of the corpora. Row 10 shows the results of SVM modified for corpora weighting (SVM-CW). Using one of the corpora as the target corpus, SVM-CW weights the remaining corpora (called

Table 2: Evaluation results. Performance is reported in terms of Recall/Precision/F-value.

Kernel	AIMed	BioInfer	HPRD50	IEPA	LLL
1 (Tikk et al., 2010)	53.6/59.9/56.2	61.3/60.2/60.7	69.8/68.2/67.8	82.6/66.6/**73.1**	98.0/68.0/78.4
2 (Miwa et al., 2009a)	68.8/55.0/60.8	71.1/65.7/**68.1**	76.1/68.5/70.9	78.6/67.5/71.7	86.0/77.6/80.1
Edit kernel					
3 SDG	40.0/61.4/48.4	64.7/49.5/56.1	55.8/68.4/61.5	69.6/74.7/72.0	89.6/71.7/79.7
4 EDG	57.3/65.3/**61.1**	57.6/59.9/58.7	66.9/75.7/71.0	69.9/76.2/72.9	85.4/74.1/79.3
5 EDG (Best)	–	–	76.7/83.3/**79.9**	–	92.1/78.2/**84.6**
All-paths graph kernel					
6 SDG	69.0/48.0/56.6	73.5/58.8/65.3	69.3/60.1/64.4	77.9/65.4/71.1	87.8/69.9/77.8
7 EDG	66.0/52.3/58.3	72.1/56.1/63.1	71.2/62.7/66.7	75.2/65.3/69.9	82.9/69.4/75.6
8 EDG (Best)	71.3/51.1/59.5	69.2/58.7/63.5	76.1/62.6/68.7	76.1/68.2/71.9	87.2/75.3/80.8
Feature vector (Miwa et al., 2009b)					
9 L2-SVM	63.2	66.2	67.2	73.0	80.3
10 SVM-CW	64.0	66.7	72.7	75.2	85.9

the source corpora) with "goodness" for training on the target corpus, adjusting the effect of their compatibility and incompatibility (Miwa et al., 2009b). Thus, their results are not directly comparable with our results. However we obtain improvements using edit kernel with EDG (Best) on HPRD50.

4.4 Contribution of individual relation

Table 3 compares the effects of different techniques in EDG on five corpora using the edit distance kernel. We first evaluated SDG obtained from the Stanford conversion tool with "CCProcessed" setting (Row 2) for processing conjunctions, and next added only syntax based *arg0* and *arg1* (Row 3). After that, we added in succession referential links (including coreference, appositive, and *is-a*), *member-collection*, and *part-whole* detection in the EDG construction step by step (Row 4–6). Overall, using "CCProcessed" increases the F-values on all five corpora. EDG constructed using syntax based arg achieves additional increases on 4 out of 5 corpora (exception was IEPA). Every subsequent step generally provides more improvements on F-values. However, we observed that on HPRD50, *member-collection* decreased F-value. Therefore we tried to switch off this part in the EDG construction but included the rest of the relations and achieved a higher F-value of 79.9% on this corpus (Row 7). This corresponds to the same result we displayed in Row 5 (EDG Best) in Table 2. On the LLL corpus, as components were successively added, we noticed a drop in F-value when referential linking was added. So similarly by turning off *is-a* detection and including all other EDG edges enabled us to obtain the EDG best F-value of 84.6% on LLL.

We also identified that *is-a* decreased F-values on IEPA, however no further improvement could be made by switching it off. We plan to further analyze this result in the future.

Additionally, due to the gap in the performance between our system and (Miwa et al., 2009a) on BioInfer, we analyzed the error cases and noticed several cases similar to the following example. The candidate pair of named enitites are marked in bold.

- This process involves other **actin-binding proteins**, such as **cofilin** and coronin.

Using techniques as shown in Figure 3, we can create *arg0 (binding, actin)* and *arg1 (binding, proteins)* in EDG and also detect *member-collection* relation between "actin-binding proteins" and "cofilin". With propogation, an interaction between "actin" and "cofilin" can be predicted. However, this relation is annotated as a negative, but instead the annotation in BioInfer includes a positive relation between "actin-binding proteins" and "cofilin". Because of similar examples in BioInfer, the *member-collection* and *is-a* and propagation failed to improve the results in BioInfer.

5 Conclusion

In this paper, we strive to find a level of abstraction that is more suitable for tasks such as relation extraction. For this purpose, we introduced techniques to create a new dependency graph representation (EDG) that goes beyond syntactic dependencies. We evaluated the efficacy of EDG

Table 3: Contributions of different part in SDG and EDG using edit kernel. Performance is reported in terms of Recall/Precision/F-value.

Kernel	AIMed	BioInfer	HPRD50	IEPA	LLL
1 SDG (Collapsed)	40.0/61.4/48.4	64.7/49.5/56.1	55.8/68.4/61.5	69.6/74.7/72.0	89.6/71.7/79.7
2 SDG (CCProcessed)	46.4/58.9/51.9	56.2/57.1/56.6	58.9/67.6/63.0	70.2/74.8/72.4	89.6/73.5/80.8
3 EDG (syntax based *arg*)	48.1/61.2/53.9	56.3/58.5/57.4	66.9/73.2/69.9	69.3/74.4/71.7	89.0/74.1/80.9
4 EDG (above, coref, app, isa)	52.2/58.6/55.2	56.7/58.3/57.5	65.6/77.0/70.9	69.0/74.0/71.4	87.2/72.2/79.0
5 EDG (above, mem-coll)	53.2/59.2/56.0	57.1/58.6/57.8	64.4/77.8/70.5	69.6/76.4/72.8	85.4/74.5/79.6
6 EDG (above, part-whole)	57.3/65.3/61.1	57.6/59.9/58.7	66.9/75.7/71.0	69.9/76.2/72.9	85.4/74.1/79.3
7 EDG (Best)	57.3/65.3/61.1	57.6/59.9/58.7	76.7/83.3/79.9	69.9/76.2/72.9	92.1/78.2/84.6

with the edit distance and APG kernels and applied them on 5 different PPI-related datasets. We obtained improvements in F-value by using EDG. We find that despite the simplicity of the edit kernel and its weak performance with the baseline graph, results comparable to state-of-the-art systems using kernel methods are obtained on different corpora with the inclusion of EDG.

While the use of EDG has led to gain in recall as well as precision mostly, the recall drops with BioInfer dataset. We would like to analyze this result further in the future. One of our main motivations for developing EDG is to develop methods to learn with small datasets and whether the abstraction captured in EDG allows for easier generalization. The testing of learning with small datasets and use in context of active learning will be investigated in the future.

We plan to test the use of EDG on other relation extraction tasks in the biomedical domain. We also plan to investigate richer features and their combinations in conjunction with the use of EDG.

Acknowledgments

Research reported in this manuscript is supported by the National Science Foundation under Grant No. DBI-1062520.

References

Antti Airola, Sampo Pyysalo, Jari Björne, 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.

Claire Bonial, Jena Hwang, Julia Bonn, Kathryn Conger, Olga Babko-Malaya, and Martha Palmer, 2012. *English PropBank Annotation Guidelines*. Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT-EMNLP*, pages 724–731, Stroudsburg, PA, USA.

Razvan C. Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. 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.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics.

Sung-Pil Choi and Sung-Hyon Myaeng. 2010. Simplicity is better: revisiting single kernel ppi extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 206–214. Association for Computational Linguistics.

Faisal Md. Chowdhury, Alberto Lavelli, and Alessandro Moschitti. 2011. 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. Association for Computational Linguistics.

Michael A. Covington. 2001. A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics.

Jing Ding, Daniel Berleant, Dan Nettleton, and E Wurtele. 2002. Mining MEDLINE: Abstracts,

sentences, or phrases. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 7, pages 326–337.

Günes Erkan, Arzucan Özgür, and Dragomir R Radev. 2007. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *EMNLP-CoNLL*, volume 7, pages 228–237.

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

Thomas Gärtner, Peter Flach, and Stefan Wrobel. 2003. On graph kernels: Hardness results and efficient alternatives. In *Conference on Learning Theory*, pages 129–143.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING*, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, Alfonso Valencia, et al. 2008. Overview of the protein-protein interaction annotation extraction task of biocreative ii. *Genome biology*, 9(Suppl 2):S4.

Haibin Liu, Karin Verspoor, Donald C Comeau, Andrew MacKinlay, and W John Wilbur. 2013. Generalizing an approximate subgraph matching-based system to extract events in molecular biology and cancer genetics. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 76–85.

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(12):e39–46.

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 the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 121–130. Association for Computational Linguistics.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 788–796, Stroudsburg, PA, USA. Association for Computational Linguistics.

Claire Nédellec. 2005. Learning language in logic - genic interaction extraction challenge. In *Proceedings of the 4th Learning Language in Logic Workshop*, volume 7, pages 31–37.

Nhung TH Nguyen, Makoto Miwa, Yoshimasa Tsuruoka, Takashi Chikayama, and Satoshi Tojo. 2015. Wide-coverage relation extraction from medline using deep syntax. *BMC bioinformatics*, 16(1):107.

Xia Ning and Yanjun Qi. 2011. Semi-supervised convolution graph kernels for relation extraction. In *SDM*, pages 510–521. SIAM.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 Task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.

Yifan Peng, Manabu Torii, Cathy H. Wu, and K. Vijay-Shanker. 2014. An NLP framework for fast development of pattern-based biomedical relation extraction systems. *BMC bioinformatics*, 15:285.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, 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.

Long Qiu, Min yen Kan, and Tat seng Chua. 2004. A public reference implementation of the rap anaphora resolution algorithm. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 291–294.

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Michael Hess, and Martin Romacker. 2006. An environment for relation mining over richly annotated corpora: the case of genia. *BMC bioinformatics*, 7(Suppl 3):S3.

A.S. Schwartz and M.A. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, pages 451–462.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17*, pages 1297–1304, Cambridge, MA. MIT Press.

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):e1000837.

Sofie Van Landeghem, Sampo Pyysalo, Tomoko Ohta, and Yves Van de Peer. 2010. Integration of static relations to enhance event extraction from text. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, BioNLP '10, pages

144–152, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sofie Van Landeghem, Jari Björne, Thomas Abeel, Bernard De Baets, Tapio Salakoski, and Yves Van de Peer. 2012. Semantically linking molecular entities in literature through entity relationships. *BMC bioinformatics*, 13(Suppl 11):S6.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 825–832, Stroudsburg, PA, USA.

Yijia Zhang, Hongfei Lin, Zhihao Yang, Jian Wang, and Yanpeng Li. 2012. Hash subgraph pairwise kernel for protein-protein interaction extraction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(4):1190–1202.

Guodong Zhou, Min Zhang, Dong Hong, and Ji Qiaoming Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *EMNLP-CoNLL*, pages 728–736.