Dependency-Driven Feature-based Learning for Extracting Protein-Protein Interactions from Biomedical Text

Bing Liu   **Longhua Qian**[*]   **Hongling Wang**   **Guodong Zhou**
Jiangsu Provincial Key Lab for Computer Information Processing Technology
School of Computer Science and Technology
Soochow University
Email: liubingnlp@gmail.com
{qianlonghua,redleaf,gdzhou}@suda.edu.cn

Abstract

Recent kernel-based PPI extraction systems achieve promising performance because of their capability to capture structural syntactic information, but at the expense of computational complexity. This paper incorporates dependency information as well as other lexical and syntactic knowledge in a feature-based framework. Our motivation is that, considering the large amount of biomedical literature being archived daily, feature-based methods with comparable performance are more suitable for practical applications. Additionally, we explore the difference of lexical characteristics between biomedical and newswire domains. Experimental evaluation on the AIMed corpus shows that our system achieves comparable performance of 54.7 in F1-Score with other state-of-the-art PPI extraction systems, yet the best performance among all the feature-based ones.

1 Introduction

In recent years, automatically extracting biomedical information has been the subject of significant research efforts due to the rapid growth in biomedical development and discovery. A wide concern is how to characterize protein interaction partners since it is crucial to understand not only the functional role of individual proteins but also the organization of the entire biological process. However, manual collection of relevant Protein-Protein Interaction (PPI) information from thousands of research papers published every day is so time-consuming that automatic extraction approaches with the help of Natural Language Processing (NLP) techniques become necessary.

Various machine learning approaches for relation extraction have been applied to the biomedical domain, which can be classified into two categories: feature-based methods (Mitsumori et al., 2006; Giuliano et al., 2006; Sætre et al., 2007) and kernel-based methods (Bunescu et al., 2005; Erkan et al., 2007; Airola et al., 2008; Kim et al., 2010).

Provided a large-scale manually annotated corpus, the task of PPI extraction can be formulated as a classification problem. Typically, for featured-based learning each protein pair is represented as a vector whose features are extracted from the sentence involving two protein names. Early studies identify the existence of protein interactions by using "bag-of-words" features (usually uni-gram or bi-gram) around the protein names as well as various kinds of shallow linguistic information, such as POS tag, lemma and orthographical features. However, these systems do not achieve promising results since they disregard any syntactic or semantic information altogether, which are very useful for the task of relation extraction in the newswire domain (Zhao and Grishman, 2005; Zhou et al., 2005). Furthermore, feature-based methods fail to effectively capture the structural information, which is essential to

---

[*] Corresponding author

identify the relationship between two proteins in a syntactic representation.

With the wide application of kernel-based methods to many NLP tasks, various kernels such as subsequence kernels (Bunescu and Mooney, 2005) and tree kernels (Li et al., 2008), are also applied to PPI detection.. Particularly, dependency-based kernels such as edit distance kernels (Erkan et al., 2007) and graph kernels (Airola et al., 2008; Kim et al., 2010) show some promising results for PPI extraction. This suggests that dependency information play a critical role in PPI extraction as well as in relation extraction from newswire stories (Culotta and Sorensen, 2004). In order to appreciate the advantages of both feature-based methods and kernel-based methods, composite kernels (Miyao et al., 2008; Miwa et al., 2009a; Miwa et al., 2009b) are further employed to combine structural syntactic information with flat word features and significantly improve the performance of PPI extraction. However, one critical challenge for kernel-based methods is their computation complexity, which prevents them from being widely deployed in real-world applications regarding the large amount of biomedical literature being archived everyday.

Considering the potential of dependency information for PPI extraction and the challenge of computation complexity of kernel-based methods, one may naturally ask the question: "Can the essential dependency information be maximally exploited in featured-based PPI extraction so as to enhance the performance without loss of efficiency?" "If the answer is Yes, then How?"

This paper addresses these problems, focusing on the application of dependency information to feature-based PPI extraction. Starting from a baseline system in which common lexical and syntactic features are incorporated using Support Vector Machines (SVM), we further augment the baseline with various features related to dependency information, including predicates in the dependency tree. Moreover, in order to reveal the linguistic difference between distinct domains we also compare the effects of various features on PPI extraction from biomedical texts with those on relation extraction from newswire narratives. Evaluation on the AIMed and other PPI corpora shows that our method achieves the best performance among all feature-based systems.

The rest of the paper is organized as follows. A feature-based PPI extraction baseline system is given in Section 2 while Section 3 describes our dependency-driven method. We report our experiments in Section 4, and compare our work with the related ones in Section 5. Section 6 concludes this paper and gives some future directions.

2 Feature-based PPI extraction: Baseline

For feature-based methods, PPI extraction task is re-cast as a classification problem by first transforming PPI instances into multi-dimensional vectors with various features, and then applying machine learning approaches to detect whether the potential relationship exists for a particular protein pair. In training, a feature-based classifier learning algorithm, such as SVM or MaxEnt, uses the annotated PPI instances to learn a classifier while, in testing, the learnt classifier is in turn applied to new instances to determine their PPI binary classes and thus candidate PPI instances are extracted.

As a baseline, various linguistic features, such as words, overlap, chunks, parse tree features as well as their combined ones are extracted from a sentence and formed as a vector into the feature-based learner.

1) Words

Four sets of word features are used in our system: 1) the words of both the proteins; 2) the words between the two proteins; 3) the words before M1 (the 1st protein); and 4) the words after M2 (the 2nd protein). Both the words before M1 and after M2 are classified into two bins: the first word next to the proteins and the second word next to the proteins. This means that we only consider the two words before M1 and after M2. Words features include:

- MW1: bag-of-words in M1
- MW2: bag-of-words in M2
- BWNULL: when no word in between
- BWO: other words in between except first and last words when at least three words in between
- BWM1FL: the only word before M1

- BWM1F: first word before M1
- BWM1L: second word before M1
- BWM1: first and second word before M1
- BWM2FL: the only word after M2
- BWM2F: first word after M2
- BWM2L: second word after M2
- BWM2: first and second word after M2

2) Overlap

The numbers of other protein names as well as the words that appear between two protein names are included in the overlap features. This category of features includes:

- #MB: number of other proteins in between
- #WB: number of words in between
- E-Flag: flag indicating whether the two proteins are embedded or not

3) Chunks

It is well known that chunking plays an important role in the task of relation extraction in the ACE program (Zhou et al., 2005). However, its significance in PPI extraction has not fully investigated. Here, the Stanford Parser[1] is first employed for full parsing, and then base phrase chunks are derived from full parse trees using the Perl script[2]. The chunking features usually concern about the head words of the phrases between the two proteins, which are further classified into three bins: the first phrase head in between, the last phrase head in between and other phrase heads in between. In addition, the path of phrasal labels connecting two proteins is also a common syntactic indicator of the polarity of the PPI instance, just as the path NP_VP_PP_NP in the sentence "*The ability of PROT1 to interact with the PROT2 was investigated.*" is likely to suggest the positive interaction between two proteins. These base phrase chunking features contain:

- CPHBNULL: when no phrase in between.
- CPHBFL: the only phrase head when only one phrase in between
- CPHBF: the first phrase head in between when at least two phrases in between.

- CPHBL: the last phrase head in between when at least two phrase heads in between.
- CPHBO: other phrase heads in between except first and last phrase heads when at least three phrases in between.
- CPP: path of phrase labels connecting the two entities in the chunking

Furthermore, we also generate a set of bi-gram features which combine the above chunk features except CPP with their corresponding chunk types.

4) Parse Tree

It is obvious that full pares trees encompass rich structural information of a sentence. Nevertheless, it is much harder to explore such information in featured-based methods than in kernel-based ones. Thus so far only the path connecting two protein names in the full-parse tree is considered as a parse tree feature.

- PTP: the path connecting two protein names in the full-parse tree.

Again, take the sentence "The ability of PROT1 to interact with the PROT2 was investigated." as an example, the parse path between PROT1 and PROT2 is NP_S_VP_PP_NP, which is slightly different from the CPP feature in the chunking feature set.

3 Dependency-Driven PPI Extraction

The potential of dependency information for PPI extraction lies in the fact that the dependency tree may well reveal non-local or long-range dependencies between the words within a sentence. In order to capture the necessary information inherent in the depedency tree for identifying their relationship, various kernels, such as edit distance kernel based on dependency path (Erkan et al., 2007), all-dependency-path graph kernel (Airola et al., 2008), and walk-weighted subsequence kernels (Kim et al., 2010) as well as other composite kernels (Miyao et al., 2008; Miwa et al., 2009a; Miwa et al., 2009b), have been proposed to address this problem. It's true that these methods achieve encouraging results, nevertheless, they suffer from prohibitive computation burden.

---

1 http://nlp.stanford.edu/software/lex-parser.shtml
2 http://ilk.kub.nl/~sabine/chunklink/

Thus, our solution is to fold the structural dependency information back into flat features in a feature-based framework so as to speed up the learning process while retaining comparable performance. This is what we refer to as dependency-driven PPI extraction.

First, we construct dependency trees from grammatical relations generated by the Stanford Parser. Every grammatical relation has the form of *dependent-type* (*word1*, *word2*), Where *word1* is the head word, *word2* is dependent on *word1*, and *dependent-type* denotes the pre-defined type of dependency. Then, from these grammatical relations the following features called DependenecySet1 are taken into consideration as illustrated in Figure 1:

- DP1TR: a list of words connecting PROT1 and the dependency tree root.
- DP2TR: a list of words connecting PROT2 and the dependency tree root.
- DP12DT: a list of dependency types connecting the two proteins in the dependency tree.
- DP12: a list of dependent words combined with their dependency types connecting the two proteins in the dependency tree.
- DP12S: the tuple of every word combined with its dependent type in DP12.
- DPFLAG: a boolean value indicating whether the two proteins are directly dependent on each other.

The typed dependencies produced by the Stanford Parser for the sentence "PROT1 contains a sequence motif binds to PROT2." are listed as follows:

```
nsubj(contains-2,PROT1-1)
det(motif-5, a-3)
nn(motif-5, sequence-4)
nsubj(binds-6, motif-5)
ccomp(contains-2, binds-6)
prep_to(binds-6, PROT2-8)
```

Each word in a dependency tuple is followed by its index in the original sentence, ensuring accurate positioning of the head word and dependent word. Figure 1 shows the dependency tree we construct from the above grammatical relations.
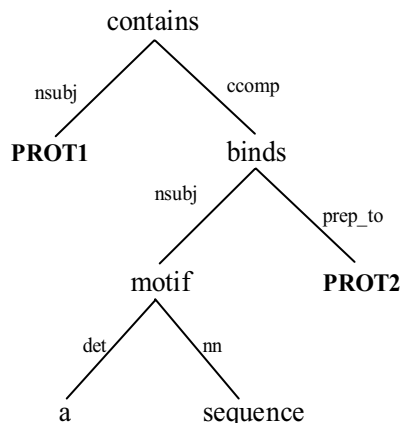

Figure 1: Dependency tree for the sentence "PROT1 contains a sequence motif binds to PROT2."

Erkan et al. (2007) extract the path information between PROT1 and PROT2 in the dependency tree for kernel-based PPI extraction and report promising results, nevertheless, such path is so specific for feature-based methods that it may incure higher precision but lower recall. Thus we alleviate this problem by collapsing the feature into multiple ones with finer granularity, leading to the features such as DP12S.

It is widely acknowledged that predicates play an important role in PPI extraction. For example, the change of a pivot predicate between two proteins may easily lead to the polarity reversal of a PPI instance. Therefore, we extract the predicates and their positions in the dependency tree as predicate features called DependencySet2:

- FVW: the predicates in the DP12 feature occurring prior to the first protein.
- LVW: the predicates in the DP12 feature occurring next to the second entity.
- MVW: other predicates in the DP12 features.
- #FVW: the number of FVW
- #LVW: the number of LVW
- #MVW: the number of MVW

4   Experimentation

This section systematically evaluates our feature-based method on the AIMed corpus as well as other commonly used corpus and reports our experimental results.

4.1 Data Sets

We use five corpora[3] with the AIMed corpus as the main experimental data, which contains 177 Medline abstracts with interactions between two interactions, and 48 abstracts without any PPI within single sentences. There are 4,084 protein references and around 1,000 annotated interactions in this data set.

For corpus pre-procession, we first rename two proteins of a pair as PROT1 and PROT2 respectively in order to blind the learner for fair comparison with other work. Then, all the instances are generated from the sentences which contain at least two proteins, that is, if a sentence contains n different proteins, there are $\binom{n}{2}$ different pairs of proteins and these pairs are considered untyped and undirected. For the purpose of comparison with previous work, all the self-interactions (59 instances) are removed, while all the PPI instances with nested protein names are retained (154 instances). Finally, 1002 positive instances and 4794 negative instances are generated and their corresponding features are extracted.

We select Support Vector Machines (SVM) as the classifier since SVM represents the state-of-the-art in the machine learning research community. In particular, we use the binary-class SVMLigh [4] developed by Joachims (1998) since it satisfies our requirement of detecting potential PPI instances.

Evaluation is done using 10-fold document-level cross-validation. Particularly, we apply the extract same 10-fold split that was used by Bunescu et al. (2005) and Giuliano et al. (2006). Furthermore, OAOD (One Answer per Occurrence in the Document) strategy is adopted, which means that the correct interaction must be extracted for each occurrence. This guarantees the maximal use of the available data, and more important, allows fair comparison with earlier relevant work.

The evaluation metrics are commonly used Precision (P), Recall (R) and harmonic F1-score (F1). As an alternative to F1-score, the AUC (*area under the receiver operating characteristics curve*) measure is proved to be invariant to the class distribution of the training dataset. Thus we also provide AUC scores

[3] http://mars.cs.utu.fi/PPICorpora/GraphKernel.html
[4] http://svmlight.joachims.org/

for our system as Airola et al. (2008) and Miwa et al. (2009a).

4.2 Results and Discussion

Features	P(%)	R(%)	F1
Baseline features			
Words	59.4	40.6	47.6
+Overlap	60.4	39.9	47.4
+Chunk	59.2	44.5	50.6
+Parse	60.9	44.8	51.4
Dependency-driven features			
+Dependency Set1	62.9	48.0	53.9
+Dependency Set2	**63.4**	**48.8**	**54.7**

Table 1: Performance of PPI extraction with various features in the AIMed corpus

We present in Table 1 the performance of our system using document-wise evaluation strategies and 10-fold cross-validation with different features in the AIMed corpus, where the plus sign before a feature means it is incrementally added to the feature set. Table 1 reports that our system achieves the best performance of 63.4/48.8/54.7 in P/R/F scores. It also shows that:

- Words features alone achieve a relatively low performance of 59.4/40.9/47.6 in P/R/F, particularly with fairly low recall score. This suggests the difficulty of PPI extraction and words features alone can't effectively capture the nature of protein interactions.

- Overlap features slightly decrease the performance. Statistics show that both the distributions of #MB and #WB between positives and negatives are so similar that they are by no means the discriminators for PPI extraction. Hence, we exclude the overlap features in the succeeding experiments.

- Chunk features significantly improves the F-measure by 3 units largely due to the increase of recall by 3.9%, though at the slight expense of precision. This suggests the effectiveness of shallow parsing information in the form of headwords captured by chunking on PPI extraction.

- The usefulness of the parse tree features is quite limited. It only improves the F-measure by 0.8 units. The main reason may be that these paths are usually long

and specific, thus they suffer from the problem of data sparsity. Furthermore, some of the parse tree features are already involved in the chunk features.

- The DependencySet1 features are very effective in that it can increase the precision and recall by 2.0 and 3.2 units respectively, leading to the increase of F1 score by 2.5 units. This means that the dependency-related features can effectively retrieve more PPI instances without introducing noise that will severely harm the precision. According to our statistics, there are over 60% sentences with more than 5 words between their protein entities in the AIMed corpus. Therefore, dependency information exhibit great potential to PPI extraction since they can capture long-range dependencies within sentences. Take the aforementioned sentence "PROT1 contains a sequence motif binds to PROT2." as an example, although the two proteins step over a relatively long distance, the dependency path between them is concise and accurate, reflecting the essence of the interaction.
- The predicate features also contribute to the F1-score gain of 0.8 units. It is not surprising since some predicates, such as "interact", "activate" and "inhibit" etc, are strongly suggestive of the interaction polarity between two proteins.

We compare in Table 2 the performance of our system with other systems in the AIMed corpus using the same 10-fold cross validation strategy. These systems are grouped into three distinct classes: feature-based, kernel-based and composite kernels. Except for Airola et al. (2008) Miwa et al. (2009a) and Kim et al. (2010), which adopt graph kernels, our system performs comparably with other systems. In particular, our dependency-driven system achieves the best F1-score of 54.7 among all feature-based systems.

In order to measure the generalization ability of our dependency-driven PPI extraction system across different corpora, we further apply our method to other four publicly available PPI corpora: BioInfer, HPRD50, IEPA and LLL.

Systems	P(%)	R(%)	F1
Feature-based methods			
Our system	63.4	48.8	54.7
Giuliano et al., 2006[5]	60.9	57.2	59.0
Sætre et al., 2007	64.3	44.1	52.0
Mitsumori et al., 2006	54.2	42.6	47.7
Yakushiji et al., 2005	33.7	33.1	33.4
Kernel-based methods			
Kim et al., 2010	61.4	53.3	56.7
Airola et al., 2008	52.9	61.8	56.4
Bunescu et al., 2006	65.0	46.4	54.2
Composite kernels			
Miwa et al., 2009a	-	-	62.0
Miyao et al., 2008[6]	51.8	58.1	54.5

Table 2: Comparison with other PPI extraction systems in the AIMed corpus

The corresponding performance of F1-score and AUC metrics as well as their standard deviations is present in Table 3. Comparative available results from Airola et al. (2008) and Miwa et al. (2009a) are also included in Table 3 for comparison. This table shows that our system performs almost consistently with the other two systems, that is, the LLL corpus gets the best performance yet with the greatest variation, while the AIMed corpus achieves the lowest performance with reasonable variation.

It is well known that biomedical texts exhibit distinct linguistic characteristics from newswire narratives, leading to dramatic performance gap between PPI extraction and relation detection in the ACE corpora. However, no previous work has ever addressed this problem and empirically characterized this difference. In this paper, we devise a series of experiments over the ACE RDC corpora using our dependency-driven feature-based method as a touchstone task. In order to do that, a sub-

---

[5] Airola et al. (2008) repeat the method published by Giuliano et al. (2006) with a correctly preprocessed AIMed and reported an F1-score of 52.4%.
[6] The results from Table 1 (Miyao et al., 2009) with the most similar settings to ours (Stanford Parser with SD representation) are reported.

set of 5796 relation instances is randomly sampled from the ACE 2003 and 2004 corpora respectively. The same cross-validation and evaluation metrics are applied to these two sets as PPI extraction in the AIMed corpus.

Corpus	Our system				Airola et al. (2008)[7]				Miwa et al. (2009a)			
	F1	σ_{F1}	AUC	σ_{AUC}	F1	σ_{F1}	AUC	σ_{AUC}	F1	σ_{F1}	AUC	σ_{AUC}
AIMed	54.7	4.5	82.4	3.5	56.4	5.0	84.8	2.3	60.8	6.6	86.8	3.3
BioInfer	59.8	3.5	80.9	3.3	61.3	5.3	81.9	6.5	68.1	3.2	85.9	4.4
HPRD50	64.9	13.4	79.8	8.5	63.4	11.4	79.7	6.3	70.9	10.3	82.2	6.3
IEPA	62.1	6.2	74.8	6.6	75.1	7.0	85.1	5.1	71.7	7.8	84.4	4.2
LLL	78.1	15.8	85.1	8.3	76.8	17.8	83.4	12.2	80.1	14.1	86.3	10.8

Table 3: Comparison of performance across the five PPI corpora

Features	AIMed			ACE2003			ACE2004		
	P(%)	R(%)	F1	P(%)	R(%)	F1	P(%)	R(%)	F1
Words	59.4	40.6	47.6	66.5	51.6	57.9	68.1	59.6	63.4
+Overlap	+1.0	-0.7	-0.2	+5.4	+1.8	+3.2	+4.6	+1.2	+2.7
+Chunk	-1.7	+4.6	+3.2	+2.3	+5.1	+4.0	+1.5	+1.9	+1.7
+Parse	+1.7	+0.3	+0.8	+0.3	+0.6	+0.5	+0.6	+0.4	+0.5
+Dependency Set1	+2.0	+3.2	+2.5	+0.8	+0.7	+0.7	+0.5	+0.9	+0.7
+Dependency Set2	+0.5	+0.8	+0.8	+0.3	+0.2	+0.3	+0.2	+0.4	+0.3

Table 4: Comparison of contributions of different features to relation detection across multiple domains

Table 4 compares the performance of our method over different domains. The table reports that the words features alone achieve the best F1-score of 63.4 in ACE2004 but the lowest F1-score of 47.6 in AIMed. This suggests the wide difference of lexical distribution between these domains. We extract the words appearing before the 1st mention, between the two mentions and after the 2nd mention from the training sets of these corpora respectively, and summarize the statistics (the number of tokens, the number of occurrences) in Table 5, where the KL divergence between positives and negatives is summed over the distribution of the 500 most frequently occurring words.

Statistics	AIMed	ACE2003	ACE2004
# of tokens	2,340	2,064	2,099
# of occurrences	69,976	53,744	49,570
KL divergence	0.22	0.28	0.33

Table 5: Lexical statistics on three corpora

The table shows that AIMed uses the most kinds of words and the most words around the two mentions than the other two. More important, AIMed has the least distribution difference between the words appearing in positives and negatives, as indicated by its least KL divergence. Therefore, the lexical words in AIMed are less discriminative for relation detection than they do in the other two. This naturally explains the reason why the performance by words feature alone is AIMed<ACE2003<ACE2004. In addition, Table 4 also shows that:

- The overlap features significantly improve the performance in ACE while slightly deteriorating that in AIMed. The reason is that, as indicated in Zhou et al. (2005), most of the positive relation instances in ACE exist in local contexts, while the positive interactions in AIMed occur in relative long-range just as the negatives, therefore these features are not discriminative for AIMed.
- The chunk features consistently greatly boost the performance across multiple corpora. This implies that the headwords in chunk phrases can well capture the partial nature of relation instances regardless of their genre.
- It's not surprising that the parse feature attain moderate performance gain in all domains since these parse paths are usually

long and specificity, leading to data sparseness problem.

- It is interesting to note that the dependency-related features exhibit more significant improvement in AIMed than that in ACE. The reason may be that, these dependency features can effectively capture long-range relationships prevailing in AIMed, while in ACE a large number of local relationships dominate the corpora.

5   Related Work

Among feature-based methods, the PreBIND system (Donaldson et al., 2003) uses words and word bi-grams features to identify the existence of protein interactions in abstracts and such information is used to enhance manual expert reviewing for the BIND database. Mitsumori et al. (2006) use SVM to extract protein-protein interactions, where bag-of-words features, specifically the words around the protein names, are employed. Sugiyama et al. (2003) extract various features from the sentences based on the verbs and nouns in the sentences such as the verbal forms, and the part-of-speech tags of the 20 words surrounding the verb. In addition to word features, Giuliano et al. (2006) extract shallow linguistic information such as POS tag, lemma, and orthographic features of tokens for PPI extraction. Unlike our dependency-driven method, these systems do not consider any syntactic information.

For kernel-based methods, there are several systems which utilize dependency information. Erkan et al. (2007) defines similarity functions based on cosine similarity and edit distance between dependency paths, and then incorporate them in SVM and KNN learning for PPI extraction. Airola et al. (2008) introduce all-dependency-paths graph kernel to capture the complex dependency relationships between lexical words and attain significant performance boost at the expense of computational complexity. Kim et al. (2010) adopt walk-weighted subsequence kernel based on dependency paths to explore various substructures such as e-walks, partial match, and non-contiguous paths. Essentially, their kernel is also a graph-based one.

For composite kernel methods, Sætre et al. (2007) combine a "bag-of-words" kernel with dependency and PAS (Predicate Argument Structure) tree kernels to exploit both the words features and the structural syntactic information. Hereafter, Miyao et al. (2008) investigate the contribution of various syntactic features using different representations from dependency parsing, phrase structure parsing and deep parsing by different parsers. Miwa et al. (2009a) integrate "bag-of-words" kernel, PAS tree kernel and all-dependency-paths graph kernel to achieve the higher performance. They (Miwa et al., 2009b) also use similar composite kernels for corpus weighting learning across multiple PPI corpora.

6   Conclusion and Future Work

In this paper, we have combined various lexical and syntactic features, particularly dependency information, into a feature-based PPI extraction system. We find that the dependency information as well as the chunk features contributes most to the performance improvement. The predicate features involved in the dependency tree can also moderately enhance the performance. Furthermore, comparative study between biomedical domain and the ACE newswire domain shows that these domains exhibit different lexical characteristics, rendering the task of PPI extraction much more difficult than that of relation detection from the ACE corpora.

In future work, we will explore more syntactic features such as PAS information for feature-based PPI extraction to further boost the performance.

References

A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction

with evaluation of cross corpus learning. *BMC Bioinformatics*.

R. Bunescu, R. Ge, R. Kate, E. Marcotte, R. Mooney, A. Ramani, and Y. Wong. 2005. Comparative Experiments on learning information extractors for Proteins and their interactions. *Journal of Artificial Intelligence In Medicine,* 33(2).

R. Bunescu and R. Mooney. 2005. Subsequence kernels for relation extraction. In *Proceedings of NIPS'05*, pages 171–178.

A. Culotta and J. Sorensen. 2004. Dependency Tree Kernels for Relation Extraction. In *Proceedings of ACL'04.*

I. Donaldson, J. Martin, B. de Bruijn, C. Wolting, V. Lay, B. Tuekam, S. Zhang, B. Baskin, G. D. Bader, K. Michalockova, T. Pawson, and C. W. V. Hogue. 2003. Prebind and textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *Journal of BMC Bioinformatics*, 4(11).

G. Erkan, A. Özgür, and D.R. Radev. 2007. Semi-Supervised Classification for Extracting Protein Interaction Sentences using Dependency Parsing, In *Proceedings of EMNLP-CoNLL'07*, pages 228–237.

C. Giuliano, A. Lavelli, and L. Romano. 2006. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. In *Proceedings of EACL'06*, pages 401–408.

S. Kim, J. Yoon, J. Yang, and S. Park. 2010. Walk-weighted subsequence kernels for protein-protein interaction extraction. *Journal of BMC Bioinformatics*, 11(107).

J. Li, Z. Zhang, X. Li, and H. Chen. 2008. Kernel-Based Learning for Biomedical Relation extraction. *Journal of the American Society for Information Science and Technology*, 59(5).

T. Mitsumori, M. Murata, Y. Fukuda, K. Doi, and H. Doi. 2006. Extracting protein-protein interaction information from biomedical text with SVM. *IEICE Transactions on Information and System,* E89-D (8)*.*

M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii. 2009a. Protein-Protein Interaction Extraction by Leveraging Multiple Kernels and Parsers. *Journal of Medical Informatics,* 78(2009).

M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii. 2009b. A Rich Feature Vector for Protein-Protein Interaction Extraction from Multiple Corpora. In *Proceedings of EMNLP'09*, pages 121–130.

Y. Miyao, R. Sætre, K. Sagae, T. Matsuzaki, and J.Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *Proceedings of ACL'08*, pages 46–54.

T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. 2001. Automated extraction of information on protein-protein interactions from the biological literature. *Journal of Bioinformatics*, 17(2).

K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. 2003. Extracting information on protein-protein interactions from biological literature based on machine learning approaches. *Journal of Genome Informatics*, (14): 699–700.

R. Sætre, K. Sagae, and J. Tsujii. 2007. Syntactic features for protein-protein interaction extraction. In *Proceedings of LBM'07*, pages 6.1–6.14.

A. Yakushiji, M. Yusuke, T. Ohta, Y. Tateishi, J. Tsujii. 2006. Automatic construction of predicate-argument structure patterns for biomedical information extraction. In *Proceedings of EMNLP'06*, pages 284–292.

S.B. Zhao and R. Grishman. 2005. Extracting Relations with Integrated Information Using Kernel Methods. In *Proceedings of ACL'05*, pages 419-426.

G.D. Zhou, J. Su, J. Zhang, and M. Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL'05*, pages 427-434.