An Evaluation of the Effect of Automatic Preprocessing on Syntactic Parsing for Biomedical Relation Extraction

Md. Faisal Mahbub Chowdhury[1,2] and **Alberto Lavelli**[1]

[1] Fondazione Bruno Kessler (FBK-irst), Italy
[2] University of Trento, Italy
chowdhury@fbk.eu lavelli@fbk.eu

Abstract

Relation extraction (RE) is an important text mining task which is the basis for further complex and advanced tasks. In state-of-the-art RE approaches, syntactic information obtained through parsing plays a crucial role. In the context of biomedical RE previous studies report usage of various automatic preprocessing techniques applied before parsing the input text. However, these studies do not specify to what extent such techniques improve RE results and to what extent they are corpus specific as well as parser specific. In this paper, we aim at addressing these issues by using various preprocessing techniques, two syntactic tree kernel based RE approaches and two different parsers on 5 widely used benchmark biomedical corpora of the protein-protein interaction (PPI) extraction task. We also provide analyses of various corpus characteristics to verify whether there are correlations between these characteristics and the RE results obtained. These analyses of corpus characteristics can be exploited to compare the 5 PPI corpora.

Keywords: relation extraction, parsing, preprocessing.

1. Introduction

Relation extraction (RE) is an important text mining task which is the basis for further complex and advanced tasks (e.g. event extraction, literature based discovery, . . .). The goal of RE is to identify all the relations of interest that hold between two (or more) entities inside a given text. For example, consider 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 (which is *true* in the example above).

Biomedical RE (henceforth, bio-RE) approaches have evolved from the exploitation of statistics about co-occurrences of entities, and the use of shallow linguistic information and patterns, to methods that take advantage of full syntactic parsing (Zweigenbaum et al., 2007). The importance of syntactic relationships among words for bio-RE is evident in the state-of-the-art techniques. This is also due to the fact that currently state-of-the-art parsing techniques achieve on biomedical texts performance not far from the best achieved on newspaper articles (McClosky, 2010; Rimell and Clark, 2009; Sagae et al., 2008).[1]

Statistical parsers obviously rely on the sentence constructions found in the corresponding treebanks used for training. One of the issues regarding treebanks is that they (practically) cannot have examples that cover all the types of linguistic constructions. Furthermore, since state-of-the-art parsers exploit probabilities, certain syntactic structures which are found only a few times in treebanks might be ignored in favor of syntactic structures which are more frequent. A further element is that natural language sentences, especially in biomedical texts, are often long and ambiguous.

The objective of various automatic preprocessing techniques[2] is to fix (as much as possible) inconsistencies or irregularities (which are not necessarily erroneous or wrong sentence constructions) inside the input sentences. In other words, the goal is to make the sentence constructions as much uniform as possible. Such preprocessing techniques could be helpful to "adapt" the sentences to the idiosyncrasies of the treebank on which the parser was trained. This could reduce the errors produced by tokenization and POS (part-of-speech) tagging, leading to the reduction of parsing errors and thus contributing to the overall performance on the target NLP task (e.g. RE) that uses a syntactic parser as a component.

Obviously, how to preprocess the input text is itself related to the target NLP task in which the parse trees will be used. Simplifying or removing some parts of the sentences as preprocessing might risk losing important information as well as distorting the originally intended meaning of the sentence. But for a task like RE, this could actually help to reduce redundant or non informative information. Because, not all the words (and not whole parse tree) are exploited by RE approaches. RE approaches usually focus on the words or the parts of the sentence which might contain possible cues.

[1] However, portability of syntactic parsing between different domains and text genres remains an open issue.

[2] For example, to simplify them, to add relevant punctuation wherever possible, to organize sentence structure in a better way, . . .

Previous studies (Bui et al., 2010; Miwa et al., 2009; Miwa et al., 2008) report usage of various preprocessing techniques for bio-RE, specifically for exploiting syntactic information for Protein-Protein Interaction (PPI) extraction, the most widely studied information extraction task in the BioNLP field. However, these studies do not explicitly address the following research questions:

1. To what extent do preprocessing techniques improve RE results?
2. To what extent are preprocessing techniques corpus specific?
3. To what extent are preprocessing techniques parser specific?[3]

Moreover, since these studies apply different preprocessing steps before exploiting different RE methodologies, it is difficult to distinguish between the contribution of the preprocessing steps and the one of the RE methodologies to the final result.

Therefore, our aim is to study the impact of preprocessing techniques based on a specific target task using the same RE methodology. In this paper we primarily focus on the first two questions above and partially to the third question. To the best of our knowledge, no such empirical study of evaluation regarding various preprocessing steps has been reported yet.

We select the PPI extraction task for our study since most of the bio-RE work are done in this context and also due to relevance of this task for understanding biological processes. Previous PPI studies that evaluated their RE systems on multiple PPI corpora showed a considerable difference among the results on different corpora even for the same system. Pyysalo et al. (2008) observed that the F1-score performance of a state-of-the-art PPI extraction method varies on average 19% when evaluated on 5 benchmark PPI corpora. Nevertheless, there has been no attempt by these studies to analyse and compare these corpora to understand why there is such performance variation.[4] Part of this study attempts to shade light on the variation of different corpus characteristics which could be used for corpora comparison.

For the evaluation on multiple corpora (which concerns the second research question mentioned above) we use the 5 benchmark PPI corpora (more in Section 4.) that are used in various PPI extraction studies. The decision of using multiple corpora is motivated by the need to evaluate whether the impact of a certain preprocessing technique is consistent across different corpora, and, if not, then to explain why it is so.

To partially address the third research question mentioned at the beginning, we exploit two widely used parsers for bio-RE: Charniak-Johnson reranking parser (henceforth, Charniak parser) (Charniak and Johnson, 2005) along with a self-trained biomedical parsing model (McClosky, 2010) and Stanford parser (Klein and Manning, 2003) (version 1.6.8). Both the parsers use the Genia (biomedical) treebank (Tateisi et al., 2005) as part of their training data. The aim is to verify our working hypothesis that preprocessing steps are parser specific.

The remainder of the paper is organized as follows. In Section 2., we briefly review previous work. Then, in Section 3., we describe the preprocessing techniques used in the experiments. Section 4. lists the datasets. Section 5. describes the empirical results. Finally, we conclude with a summary of our study.

2. Related Work

There is some earlier work on the benchmarking of natural language parsers on biomedical data (e.g. Clegg and Shepherd (2007)). However, the first work that we are aware of on evaluating contributions of parsers to a specific bio-RE task is the one performed by Miyao et al. (2009). They studied how the choice of different syntactic parsers (as a component of a PPI extraction system) and their output representations can influence bio-RE results on AIMed, a PPI corpus. Another related work is by Miwa et al. (2009). As part of their study, they evaluate their system on the AIMed corpus using two different 10-fold splittings and two preprocessing steps (tag fixes and sentence splitting). Their objective was to verify whether the choice of different corpus splittings affects the results. Since their system is composed of multiple parsers (and multiple kernels), and since they do not report the result without preprocessing, it is difficult to estimate the impact of the two preprocessing steps used.

State-of-the-art PPI extraction approaches are based on hybrid kernels[5] (Miwa et al., 2009; Chowdhury and Lavelli, 2012). These approaches use tree and graph kernels, apart from feature-based kernels, in the formation of the hybrid kernels. Since our evaluation is focused on evaluating the changes in the parser output, we use two syntactic tree kernels rather than the hybrid kernels, more precisely *(i)* the Phrase Structure Tree (PST) kernel (Moschitti, 2004)[6], and *(ii)* the Mildly Extended Dependency Tree (MEDT) kernel (Chowdhury et al., 2011). Both kernels use only syntactic information.

One of our evaluation goals includes to study the changes of the contribution of syntactic dependencies due to the

[3] Syntactic parsers can differ in approaches (e.g. lexicalized, unlexicalized, self-trained, . . .) and in methodologies (e.g. phrase structure parsing, dependency parsing, . . .).

[4] Some of these studies used cross-corpus evaluation (i.e. holding out one corpus as test set and training on the other remaining multiple corpora) without justifying whether these different corpora are similar and could be merged together for training.

[5] The term "hybrid kernel" refers to those kernels that combine multiple types of kernels.

[6] Also known as path enclosed tree (PET) kernel or shortest path-enclosed tree (SPT) kernel.

application of different preprocessing steps. Miwa et al. (2010) presented a task-oriented comparison of five parsers, measuring their contribution to bio-molecular event extraction. They used domain models with three different dependency formats – Stanford Dependency (SD), the CoNLL-X dependency and the predicate-argument structure formats. They obtained very similar performance for all the formats. Among these dependency formats, SD is arguably the most widely used format in bio-RE because of its choice to express more fine–grained relations such as apposition (Miyao et al., 2009). This format is originally proposed for extracting dependency relations useful for practical applications (de Marneffe et al., 2006) and can be obtained from the Penn Treebank-style phrase structure tree output (produced by both the Charniak-Johnson and Stanford parsers). We exploit this possibility and use the SD format during the experiments.

3. Preprocessing Techniques

We consider the following popular preprocessing techniques mentioned in literature for our experiments:

- **Entity blinding**: It refers to replacing all mentioned entity names with a place holder. For example, the sentence *"Jun mediates a physical association with the TATA box-binding protein"* would become *"ENTITY1 mediates a physical association with the ENTITY2"*. There are two versions of blind entity names:
  - place holders with all capital letters (henceforth, blind entities with all capital letters or **BEAC**), e.g. *ENTITY1*
  - capitalized place holders with mixed case letters (henceforth, blind entities with mixed case letters or **BEMC**), e.g. *Entity1*
- **Insertion of spaces at entity name boundaries** (henceforth, **IS**): This refers to the case when entity names are part of a larger token, i.e. they are subtokens and attached with some other characters (excluding comma, full-stop, semi-colon, exclamation mark and question mark). By applying IS, empty spaces are inserted before/after the boundary characters of such subtoken entity name. For example, in the following sentence *"We took advantage of previously collected data to conduct a secondary analysis of the RBP/TTR ratio"* the two entities *RBP* and *TTR* are part of a single token.
- **Removal of parenthetical comments/remarks containing no entity names** (henceforth, **RPC**): Sometimes sentences contain additional information/remarks/comments inside parentheses, more specifically between *'('* and *')'*. The objective of this preprocessing step is to remove such comments if the comments do not contain any entities that are of interest for the target RE task (i.e. proteins in case of PPI extraction). For example, in the sentence *"Insulin-induced hypoglycemia increased by 2-fold (peak vs. baseline) plasma AVP and OT levels"* such a candidate comment for removal is *(peak vs. baseline)*.

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

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

The versions of the benchmark corpora used in our experiments contain splitted sentences[7]. So, we do not need to split text into sentences (which is another type of preprocessing).

4. Data

As stated earlier, there are 5 benchmark corpora for the PPI task that are frequently used: HPRD50 (Fundel et al., 2007), IEPA (Ding et al., 2002), LLL (Nédellec, 2005), BioInfer (Pyysalo et al., 2007) and AIMed (Bunescu et al., 2005). These corpora adopt different PPI annotation formats. For a comparative evaluation Pyysalo et al. (2008) put all of them in a common format which has become the standard evaluation format for the PPI task. In our experiments, we use the versions[8] of the corpora converted to such format. Table 1 shows various statistics regarding the 5 (converted) corpora.

5. Results and Discussions

Since one of our goals is to investigate whether there is any correlation between certain corpora characteristics and specific preprocessing techniques, we measured the values of different characteristics of the 5 corpora as shown in Table 2. Figure 1 shows these values in a chart converted in log linear scale for better understanding. As the values show, the corpora have quite different characteristics. This partially explains why the empirical results reported in previous literature (as well as in this paper) show so much variation despite the fact that all the corpora are specifically annotated to facilitate PPI extraction.

Our assumption is that, among these characteristics, $AvWordPerEnt$ (avg. no. of words per entity name), $AvEntPerSen$ (avg. no. of entities per sentence) and $AvEntWordPerSen$ (avg. no. of words in all entity names per sentence) might be directly related to the performance increment/decrement because of $Entityblinding$ (i.e., $BEMC$ and $BEAC$) preprocessing. While $AvWordBetEntPair$ (avg. no. of words between each entity pair), $AvNonEntWordPerSen$ (avg. no. of words per sentence excluding entities) and $AvWordPerSen$ (avg. no. of words per sentence) might influence the outcome because of the removal of parenthetical comments/remarks having no entity names (RPC). We also

[7] We observed some sentence splitting errors in these corpora, especially in AIMed.

[8] Available from *http://mars.cs.utu.fi/PPICorpora/*

		Corpora				
Characterstitics	**Description**	**LLL**	**IEPA**	**HPRD50**	**AIMed**	**BioInfer**
AvWordBetEntPair	Avg. no. of words between each entity pair	10.46	8.89	7.11	6.92	8.44
AvWordPerEnt	Avg. no. of words per entity name	1.05	1.22	1.21	1.29	1.24
AvEntPerSen	Avg. no. of entities per sentence	3.10	2.30	2.79	3.25	4.05
AvEntWordPerSen	Avg. no. of words in (all) entity names per sent.	3.26	2.80	3.38	4.19	5.03
AvNonEntWordPerSen	Excluding entities avg. no. of words per sent.	22.57	26.07	20.93	19.71	21.93
AvWordPerSen	Avg. no. of words per sentence	25.83	28.87	24.31	23.90	26.96
%OfSubtokEnt	Percentage of subtoken entities	15.06%	12.18%	6.91%	14.23%	11.6%

Table 2: Statistics of different characteristics of the 5 benchmark PPI corpora. All sentences (in each corpus) are considered during analyses.

Preprocessing		**Preprocessed data parsed by Charniak parser**					**Preprocessed data parsed by Stanford parser**				
type		**LLL**	**IEPA**	**HPRD50**	**AIMed**	**BioInfer**	**LLL**	**IEPA**	**HPRD50**	**AIMed**	**BioInfer**
Without	**Prec.**	64.1	68.3	71.2	45.7	71.9	68.1	69.3	62.3	44.1	67.3
preprocessing	**Rec.**	90.2	76.1	63.8	64.4	72.5	84.8	72.2	69.9	57.9	69.7
(WP)	**F1**	74.9	72.0	67.3	53.5	72.2	75.5	70.8	65.9	50.1	68.5
BEMC	**Prec.**	66.7	69.9	65.5	43.9	72.1	71.0	69.1	56.7	43.0	71.3
	Rec.	91.5	77.0	69.9	73.5	75.4	92.7	75.5	85.3	67.2	70.9
	F1	**77.1**	**73.3**	**67.7**	**55.0**	**73.8**	**80.4**	**72.2**	**68.1**	**52.4**	**71.1**
BEAC	**Prec.**	67.3	72.2	56.8	43.5	72.7	72.2	71.3	55.5	42.7	70.6
	Rec.	91.5	76.7	79.1	74.3	75.0	90.2	75.5	83.4	66.1	71.7
	F1	**77.5**	**74.4**	66.2	**54.9**	**73.8**	**80.2**	**73.3**	**66.7**	**51.9**	**71.1**
IS	**Prec.**	62.0	66.4	65.5	46.4	68.8	70.8	67.6	59.4	43.6	66.8
	Rec.	92.7	74.3	67.5	64.3	72.1	93.3	72.8	71.8	61.3	70.3
	F1	74.3	70.1	66.5	**53.9**	70.4	**80.5**	70.1	65.0	**51.0**	68.5
RPC	**Prec.**	65.1	70.3	60.7	47.1	72.0	67.5	68.3	64.2	41.9	66.9
	Rec.	90.9	79.7	71.2	59.7	72.3	84.8	75.2	73.6	60.8	69.4
	F1	**75.8**	**74.7**	65.5	52.7	72.2	75.1	**71.6**	**68.6**	49.6	68.1

Table 3: Comparison of the results *using the PET kernel (constructed on the phrase structure tress)* on different preprocessed data. The 5 corpora are preprocessed separately and then each of them is parsed using either Charniak or Stanford parser. Bold **F1** score indicates performance improvement with respect to the results obtained without preprocessing.

assume that $\%OfSubtokEnt$ (percentage of subtoken entities) might help to understand the changes in the results for the insertion of spaces at entity name boundaries (IS).

Tables 3 and 4 show a comparison of the PPI extraction results without and with the preprocessing techniques. Each of the preprocessing steps is applied on the sentences before parsing them using the two parsers. We limit our study on the usage of single preprocessing technique rather than applying multiple techniques together.

Figures 2, 3, 4, 5 and 6 illustrate a graphical representation of the results on different individual corpora using different parsers and the two syntactic tree kernels.

5.1. Using phrase structure tree output of the two parsers

5.1.1. How preprocessing affects tokenization of subtoken entities

Among all the 5 corpora, AIMed and LLL have comparatively higher $\%OfDisconEnt$. Interestingly, when the IS preprocessing is applied on the corpora, the results improved only for AIMed and LLL. For AIMed this improvement is observed for both of the parsers, while for LLL only Stanford parser's output has managed to produce better results.

During analysis of the parsed data, we observed that Charniak parser is more robust on providing correct POS tags than Stanford parser. We also found that sometimes IS causes errors by parsers. For example, consider the noun phrase (NP) *"sigmaB- and sigmaF-dependent promoters"* (which actually means *"sigmaB-dependent and sigmaF-dependent promoters"*). The ideal POS and tokenization output for it should be *"sigmaB-/JJ and/CC sigmaF-/JJ dependent/JJ promoters/NNS"*. However, the parsers give the following output for this NP. Here, red text indicates tokenization error, while blue text indicates POS tagging error.

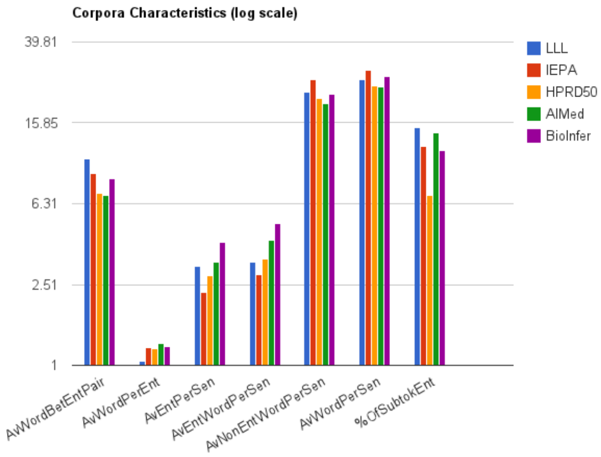

Figure 1: Chart of different corpora characteristic values in log scale.

- By Stanford parser:
  - **Without preprocessing:** *sigmaB/JJ -/: and/CC sigmaF-dependent/JJ promoters/NNS*
  - **After applying *IS*:** sigmaB/JJ -/: and/CC sigmaF/JJ -/: dependent/JJ promoters/NNS
- By Charniak parser:
  - **Without preprocessing:** sigmaB-/JJ and/CC sigmaF-dependent/JJ promoters/NNS
  - **After applying *IS*:** sigmaB/NN -/CC and/CC sigmaF/NN dependent/JJ promoters/NNS

Note that the values of $AvWordPerEnt$ and $AvEntPerSen$ are lower for LLL than for AIMed. As a result, despite IS has enabled separation of different entities (which otherwise would be part of a larger token) in separate tokens, $AvWordPerEnt$ and $AvEntPerSen$ might be not significant enough for LLL to compensate POS tagging and tokenization errors introduced by IS when Charniak parser is used.

5.1.2. When parenthetical remarks are removed

As we can see, only the results (using both parsers) on the IEPA corpus improve when RPC preprocessing is applied. We assume this is because $AvWordBetEntPair * AvNonEntWordPerSen$ for IEPA is considerably higher than for other corpora.

5.1.3. When entities are blinded

As the results show, even minor differences in seemingly the same preprocessing can have different impact on the outcome. For example, for *entity blinding* technique, the experiments show that the usage of $BEAC$ (e.g. *ENTITY1*) and $BEMC$ (e.g. *Entity1*) produces different results.

After checking the parsed data we have found in most cases that Stanford parser attaches the POS tag NNS (noun, common, plural) to tokens like *"Entity1"* and NN (noun, common, singular or mass) to tokens like *"ENTITY1"*. We have also noticed that Stanford parser can do better NP identification when $BEMC$ is used instead of $BEAC$. For example, consider the following sentence preprocessed using $BEMC$:

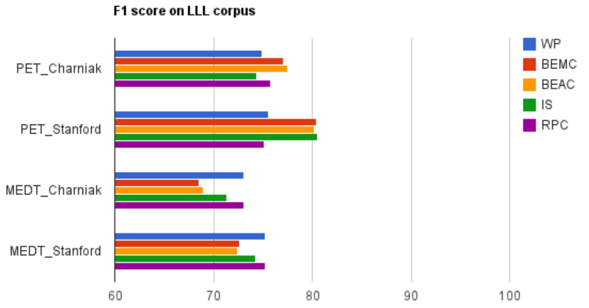

Figure 2: Graphical representation of results on LLL corpus.

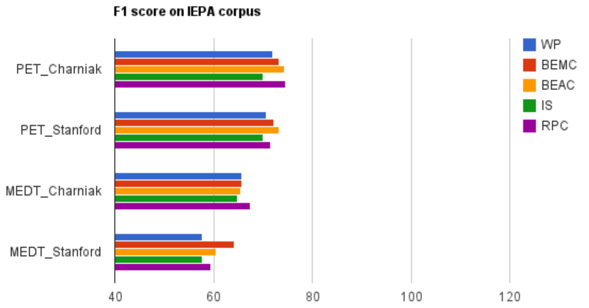

Figure 3: Graphical representation of results on IEPA corpus.

> "Cotransfection of Entity0 and Entity1 in embryonic kidney 293 cells activated the anti-apoptotic transcription factor Entity2."

Stanford parser correctly recognizes the NP *"Entity0 and Entity1 in embryonic kidney 293 cells"* (which forms a larger NP with *"Cotransfection of"*). However, if BEAC is adopted as a preprocessing step instead of BEMC, the parser fails to recognize the NP *"ENTITY0 and ENTITY1 in embryonic kidney 293 cells"* as a whole.

In the case of Charniak parser, we have observed that in most cases it attaches either NNP (noun, proper, singular) or NN tag to tokens like *"Entity1"*, and NN to tokens like *"ENTITY1"*. Like Stanford parser, better NP identification by Charniak parser is noticed when $BEMC$ is used instead of $BEAC$. For example, consider the following sentence preprocessed using $BEMC$:

> "We tested this point by cotransfecting CHO cells with the genes encoding F beta alpha and

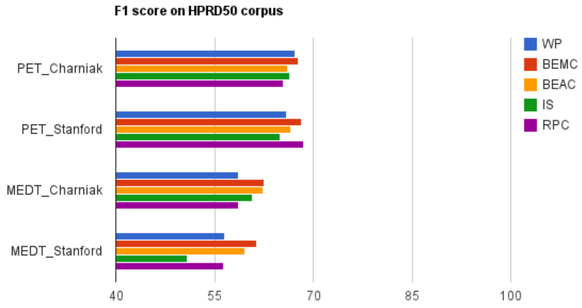

Figure 4: Graphical representation of results on HPRD50 corpus.

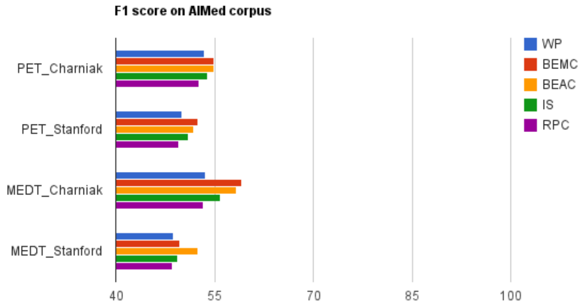

Figure 5: Graphical representation of results on AIMed corpus.

> the Entity0 subunit or the Entity1 and Entity2 monomer."

Charniak parser correctly recognizes that *"the Entity0 subunit or the Entity1 and Entity2 monomer"* is an NP which later forms a larger NP with *"F beta alpha and"*. However, if $BEAC$ is applied instead of $BEMC$, the parser wrongly identifies *"F beta alpha and the ENTITY0 subunit"* as an NP and then forms the larger NP with *"or the ENTITY1 and ENTITY2 monomer"*.

In addition, we also observe that:

- In all of the corpora, using both parsers, $BEMC$ preprocessing produced improved results (with respect to the results doing no preprocessing). An almost similar trend is seen for $BEAC$ except that the usage of the output of Charniak parser on HPRD50 produced lower results.

- Interestingly, regardless of the corpus and of the blinding technique used (i.e. $BEAC$ or $BEMC$), improvement of results (with respect to the results when no preprocessing is done) using Stanford parser output is always higher than that using Charniak parser output.

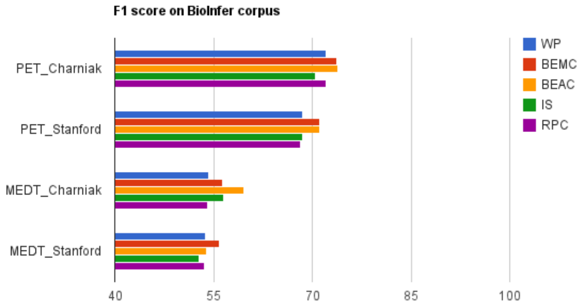

Figure 6: Graphical representation of results on BioInfer corpus.

- It is also noticeable that, without preprocessing, results are always better when Charniak parser output is exploited rather than Stanford parser output. But when entity blinding is applied the results using Stanford parser either gets better (in case of LLL and HPRD50) or closer to that using Charniak parser.

5.2. Using dependency trees obtained through the syntax tree output of the two parsers

As mentioned in Section 2., previous studies show that different dependency formats provide very similar performance. SD is arguably the most widely used dependency format in bio-RE. So, we used the SD format (which are obtained from the phrase structure trees) during our experiments with the MEDT kernel.

Empirical results show that the application, before parsing, of a preprocessing technique which might have improved the results when phrase structure trees are used, does not necessarily guarantee that the exploitation of the dependency trees derived from those phrase structure trees would lead to a better result as well.

However, we note that there are significant improvements of the results on BioInfer, AIMed and HPRD50 corpora when $BEMC$, $BEAC$ and IS are used before parsing with Charniak parser.

5.3. Other observations

As the empirical outcomes show, results on AIMed and HPRD50 are much lower than on the other corpora. Perhaps, this can be partly explained by the fact that these two corpora have lower values for $AvWordBetEntPair$, $AvNonEntWordPerSen$ and $AvWordPerSen$. However, HPRD50 has less $AvEntPerSen$ than AIMed which might have enabled to obtain slightly better results on HPRD50.

Preprocessing		**Preprocessed data parsed by Charniak parser**					**Preprocessed data parsed by Stanford parser**				
type		**LLL**	**IEPA**	**HPRD50**	**AIMed**	**BioInfer**	**LLL**	**IEPA**	**HPRD50**	**AIMed**	**BioInfer**
Without preprocessing (WP)	**Prec.**	62.6	67.4	56.6	50.2	44.5	60.3	55.0	53.9	42.6	47.6
	Rec.	87.8	64.2	60.7	57.4	69.7	100.0	60.9	59.5	57.2	61.8
	F1	73.1	65.8	58.6	53.6	54.3	75.2	57.8	56.6	48.8	53.8
BEMC	**Prec.**	52.8	62.0	60.7	58.1	46.9	58.0	60.0	58.7	45.8	50.1
	Rec.	97.6	70.2	64.4	60.3	70.6	97.0	69.3	64.4	54.5	63.3
	F1	68.5	65.8	**62.5**	**59.2**	**56.4**	72.6	**64.3**	**61.4**	**49.8**	**56.0**
BEAC	**Prec.**	53.7	63.5	60.3	55.4	52.2	57.2	54.7	54.9	51.8	45.2
	Rec.	96.3	67.5	64.4	61.7	69.5	98.8	67.8	65.0	53.1	67.0
	F1	69.0	65.4	**62.3**	**58.4**	**59.6**	72.5	**60.5**	**59.6**	**52.4**	**54.0**
IS	**Prec.**	60.7	63.2	58.1	50.5	51.9	59.6	53.8	48.1	44.8	46.0
	Rec.	86.6	66.6	63.8	62.5	62.0	98.8	62.4	54.0	55.2	62.1
	F1	71.4	64.8	**60.8**	**55.9**	**56.5**	74.3	57.8	50.9	**49.4**	52.9
RPC	**Prec.**	62.6	68.2	56.6	50.3	44.7	60.3	57.4	53.0	43.0	47.9
	Rec.	87.8	66.6	60.7	56.7	68.4	100.0	61.5	60.1	55.7	60.8
	F1	73.1	**67.4**	58.6	53.3	54.1	75.2	**59.4**	56.3	48.5	53.6

Table 4: Comparison of the results *using the MEDT kernel (constructed on the dependency graphs obtained using the phrase structure tress)* on different preprocessed data. The 5 corpora are preprocessed separately and then each of them is parsed using either Charniak or Stanford parser. Bold **F1** score indicates performance improvement with respect to the results obtained without preprocessing.

6. Conclusion

In this paper, we have presented an empirical evaluation of various preprocessing techniques (before parsing) and their contribution in a specific bio-RE task, i.e. PPI extraction, using a phrase structure tree based and a dependency tree based syntactic kernels. We have also provided analyses of different corpora characteristics of the 5 corpora used in our experiments. Based on the empirical results of our study and analyses of the output of two different parsers, we have made an attempt to relate different characteristics of a corpus with the different preprocessing techniques used. We have provided supporting examples to show how the preprocessing techniques affect output of the parsers.

We speculate that some of the preprocessing steps might complement some of the other steps and would produce larger gains in results if they are combined. However, in this paper we have limited our study to the contribution of individual techniques. Empirical results show some interesting findings, e.g. even apparently minor differences in seemingly same preprocessing can have different impact on the outcome. Also, it turns out that having an improvement (due to preprocessing) using phrase structure trees do not necessarily implies that an improvement would be also obtained using dependency trees. Finally, we present analyses of different corpora characteristics which can be exploited to compare the 5 PPI corpora.

7. Acknowledgements

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 Matteo Negri and the three anonymous reviewers for reading the initial extended abstract of the paper and providing feedback.

8. References

Q. Bui, S. Katrenko, and P.M.A. Sloot. 2010. A hybrid approach to extract protein-protein interactions. *Bioinformatics*.

R. Bunescu, R. Ge, R.J. Kate, E.M. Marcotte, R.J. Mooney, A.K. Ramani, and Y.W. Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

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

M.F.M. Chowdhury and A. Lavelli. 2012. Combining tree structures, flat features and patterns for biomedical relation extraction. In *Proceedings of EACL 2012*.

M.F.M. Chowdhury, A. Lavelli, and A. Moschitti. 2011. A study on dependency tree kernels for automatic extraction of protein-protein interaction. In *Proceedings of ACL 2011 BioNLP Workshop*.

A. Clegg and A. Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8(1).

M.-C. de Marneffe, B. MacCartney, and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*.

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

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

D. Klein and C.D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL 2003*.

D. McClosky. 2010. *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. Ph.D.

thesis, Department of Computer Science, Brown University.

M. Miwa, R. Sætre, Y. Miyao, T. Ohta, and J. Tsujii. 2008. Combining multiple layers of syntactic information for protein-protein interaction extraction. In *Proceedings of SMBM 2008*, pages 101–108.

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.

M. Miwa, S. Pyysalo, T. Hara, and J. Tsujii. 2010. A comparative study of syntactic parsers for event extraction. In *Proceedings of ACL 2010 BioNLP Workshop*, pages 37–45.

Y. Miyao, K. Sagae, R. Sætre, T. Matsuzaki, and J. Tsujii. 2009. Evaluating contributions of natural language parsers to protein–protein interaction extraction. *Bioinformatics*, 25(3).

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

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

S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Jarvinen, and T. Salakoski. 2007. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1):50.

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.

L. Rimell and S. Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of Biomedical Informatics*, 42(5).

K. Sagae, Y. Miyao, and J. Tsujii. 2008. Challenges in mapping of syntactic representations for framework-independent parser evaluation. In *Proceedings of the Workshop on Automated Syntatic Annotations for Interoperable Language Resources*.

Y. Tateisi, A. Yakushiji, T. Ohta, and J. Tsujii. 2005. Syntax annotation for the GENIA corpus. In *Proceedings of IJCNLP 2005*, pages 222–227.

P Zweigenbaum, D Demner-Fushman, H Yu, and KB Cohen. 2007. Frontiers of biomedical text mining: current progress. *Brief Bioinform*, 8(5):358–375.