Entity-Focused Sentence Simplification for Relation Extraction

Makoto Miwa[1] **Rune Sætre**[1] **Yusuke Miyao**[2] **Jun'ichi Tsujii**[1,3,4]
[1]Department of Computer Science, The University of Tokyo
[2]National Institute of Informatics
[3]School of Computer Science, University of Manchester
[4]National Center for Text Mining
mmiwa@is.s.u-tokyo.ac.jp,rune.saetre@is.s.u-tokyo.ac.jp,
yusuke@nii.ac.jp,tsujii@is.s.u-tokyo.ac.jp

Abstract

Relations between entities in text have been widely researched in the natural language processing and information-extraction communities. The region connecting a pair of entities (in a parsed sentence) is often used to construct kernels or feature vectors that can recognize and extract interesting relations. Such regions are useful, but they can also incorporate unnecessary distracting information. In this paper, we propose a rule-based method to remove the information that is unnecessary for relation extraction. Protein–protein interaction (PPI) is used as an example relation extraction problem. A dozen simple rules are defined on output from a deep parser. Each rule specifically examines the entities in one target interaction pair. These simple rules were tested using several PPI corpora. The PPI extraction performance was improved on all the PPI corpora.

1 Introduction

Relation extraction (RE) is the task of finding a relevant semantic relation between two given target entities in a sentence (Sarawagi, 2008). Some example relation types are person–organization relations (Doddington et al., 2004), protein–protein interactions (PPI), and disease–gene associations (DGA) (Chun et al., 2006). Among the possible RE tasks, we chose the PPI extraction problem. PPI extraction is a major RE task; around 10 corpora have been published for training and evaluation of PPI extraction systems.

Recently, machine-learning methods, boosted by NLP techniques, have proved to be effective for RE. These methods are usually intended to highlight or select the relation-related regions in parsed sentences using feature vectors or kernels. The shortest paths between a pair of entities (Bunescu and Mooney, 2005) or pair-enclosed trees (Zhang et al., 2006) are widely used as focus regions. These regions are useful, but they can include unnecessary sub-paths such as appositions, which cause noisy features.

In this paper, we propose a method to remove information that is deemed unnecessary for RE. Instead of selecting the whole region between a target pair, the target sentence is simplified into simpler, pair-related, sentences using general, task-independent, rules. By addressing particularly the target entities, the rules do not affect important relation-related expressions between the target entities. We show how rules of two groups can be easily defined using the analytical capability of a deep parser with specific examination of the target entities. Rules of the first group can replace a sentence with a simpler sentence, still including the two target entities. The other group of rules can replace a large region (phrase) representing one target entity, with just a simple mention of that target entity. With only a dozen simple rules, we show that we can solve several simple well-known problems in RE, and that we can improve the performance of RE on all corpora in our PPI test-set.

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 788–796,
Beijing, August 2010

2 Related Works

The general paths, such as the shortest path or pair-enclosed trees (Section 1), can only cover a part of the necessary information for relation extraction. Recent machine-learning methods specifically examine how to extract the missing information without adding too much noise. To find more representative regions, some information from outside the original regions must be included. Several tree kernels have been proposed to extract such regions from the parse structure (Zhang et al., 2006). Also the graph kernel method emphasizes internal paths without ignoring outside information (Airola et al., 2008). Composite kernels have been used to combine original information with outside information (Zhang et al., 2006; Miwa et al., 2009).

The approaches described above are useful, but they can include unnecessary information that distracts learning. Jonnalagadda and Gonzalez (2009) applied bioSimplify to relation extraction. BioSimplify is developed to improve their link grammar parser by simplifying the target sentence in a general manner, so their method might remove important information for a given target relation. For example, they might accidentally simplify a noun phrase that is needed to extract the relation. Still, they improved overall PPI extraction recall using such simplifications.

To remove unnecessary information from a sentence, some works have addressed sentence simplification by iteratively removing unnecessary phrases. Most of this work is not task-specific; it is intended to compress all information in a target sentence into a few words (Dorr et al., 2003; Vanderwende et al., 2007). Among them, Vickrey and Koller (2008) applied sentence simplification to semantic role labeling. With retaining all arguments of a verb, Vickrey simplified the sentence by removing some information outside of the verb and arguments.

3 Entity-Focused Sentence Simplification

We simplify a target sentence using simple rules applicable to the output of a deep parser called Mogura (Matsuzaki et al., 2007), to remove noisy information for relation extraction. Our method relies on the deep parser; the rules depend on the Head-driven Phrase Structure Grammar (HPSG) used by Mogura, and all the rules are written for the parser Enju XML output format. The deep parser can produce deep syntactic and semantic information, so we can define generally applicable comprehensive rules on HPSG with specific examination of the entities.

For sentence simplification in relation extraction, the meaning of the target sentence itself is less important than maintaining the truth-value of the relation (interact or not). For that purpose, we define rules of two groups: clause-selection rules and entity-phrase rules. A clause-selection rule constructs a simpler sentence (still including both target entities) by removing noisy information before and after the relevant clause. An entity-phrase rule simplifies an entity-containing region without changing the truth-value of the relation. By addressing the target entities particularly, we can define rules for many applications, and we can simplify target sentences with less danger of losing relation-related mentions. The rules are summarized in Table 1.

Our method is different from the sentence simplification in other systems (ref. Section 2). First, our method relies on the parser, while bioSimplify by Jonnalagadda and Gonzalez (2009) is developed for the improvement of their parser. Second, our method tries to keep only the relation-related regions, unlike other general systems including bioSimplify which tried to keep all information in a sentence. Third, our entity-phrase rules modify only the entity-containing phrases, while Vickrey and Koller (2008) tries to remove all information outside of the target verb and arguments.

3.1 Clause-selection Rules

In compound or complex sentences, it is natural to assume that one clause includes both the target entities and the relation-related information. It can also be assumed that the remaining sentence parts, outside the clause, contain less related (or noisy) information. The clause-selection rules simplify a sentence by retaining only the clause that includes the target entities (and by discarding the remainder of the sentence). We define three types of

Rule Group	Rule Type	#	Example (original → simplified)
Clause Selection	Sentence Clause	1	We show that A interacts with B. → A interacts with B.
	Relative Clause	2	... A that interacts with B. → A interacts with B.
	Copula	1	A is a protein that interacts with B. → A interacts with B.
Entity Phrase	Apposition	2	a protein, A → A
	Exemplification	4	proteins, such as A → A
	Parentheses	2	a protein (A) → A
	Coordination	3	protein and A → A

Table 1: Rules for Sentence Simplification. (# is the rule count. A and B are the target entities.)

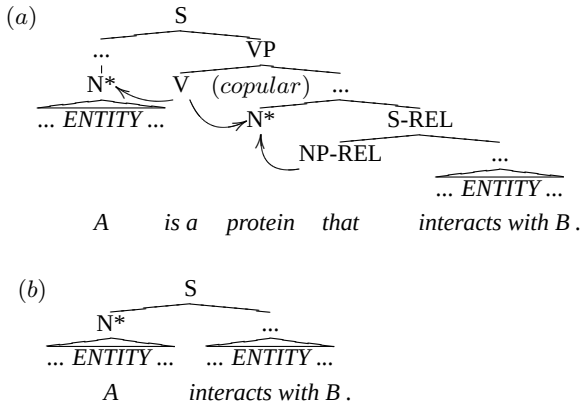

Figure 1: Copula Rule. (a) is simplified to (b). The arrows represent predicate–argument relations.

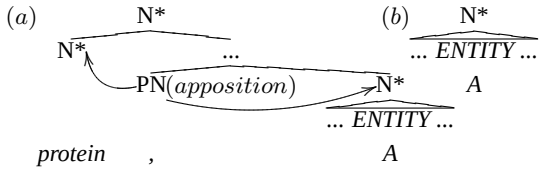

Figure 2: Apposition Rule.

clause-selection rules for sentence clauses, relative clauses, and copula. The *sentence clause rule* finds the (smallest) clause that includes both target entities. It then replaces the original sentence with the clause. The *relative clause rules* construct a simple sentence from a relative clause and the antecedent. If this simple sentence includes the target entities, it is used instead of the original sentence. We define two rules for the case where the antecedent is the subject of the relative clause. One rule is used when the relative clause includes both the target entities. The other rule is used when the antecedent includes one target entity and the relative clause includes the other target entity. The *copula rule* is for sentences that include copular verbs (e.g. be, is, become, etc). The rule constructs a simple sentence from a relative clause with the subject of the copular verb as the antecedent subject of the clause. The rule replaces the target sentence with the constructed sentence, if the relative clause includes one target entity and the subject of a copular verb includes the other target entity, as shown in Figure 1.

3.2 Entity-phrase Rules

Even the simple clauses (or paths between two target entities) include redundant or noisy expressions that can distract relation extraction. Some of these expressions are related to the target entities, but because they do not affect the truth-value of the relation, they can be deleted to make the path simple and clear. The target problem affects which expressions can be removed. We define four types of rules for appositions, exemplifications, parentheses, and coordinations. Two *apposition rules* are defined to select the correct element from an appositional expression. One element modifies or defines the other element in apposition, but the two elements represent the same information from the viewpoint of PPI. If the target entity is in one of these elements, removing the other element does not affect the truth-value of the interaction. Many of these apposition expressions are identified by the deep parser. The rule to select the last element is presented in Figure 2. Four *exemplification rules* are defined for the two major types of expressions using the phrases "including" or "such as". Exemplification is represented by hyponymy or hypernymy. As for appositions, the truth-value of the interaction does not change whether we use the specific mention or the hyperclass that the mention represents. Two *parentheses rules* are defined. Parentheses are useful for synonyms, hyponyms, or hypernyms (ref. the two

```
1: S ← input sentence
2: repeat
3:     reset rules {apply all the rules again}
4:     P ← parse S
5:     repeat
6:         r ← next rule {null if no more rules}
7:         if r is applicable to P then
8:             P ← apply r to P
9:             S ← sentence extracted from P
10:            break (Goto 3)
11:        end if
12:    until r is null
13: until r is null
14: return S
```

Figure 3: Pseudo-code for sentence simplification.

former rules). Three *coordination rules* are defined. Removing other phrases from coordinated expressions that include a target entity does not affect the truth-value of the target relation. Two rules are defined for simple coordination between two phrases (e.g. select left or right phrase), and one rule is defined to (recursively) remove one element from lists of more than two coordinated phrases (while maintaining the coordinating conjunction, e.g. "and").

3.3 Sentence Simplification

To simplify a sentence, we apply rules repeatedly until no more applications are possible as presented in Figure 3. After one application of one rule, the simplified sentence is re-parsed before attempting to apply all the rules again. This is because we require a consistent parse tree as a starting point for additional applications of the rules, and because a parser can produce more reliable output for a partly simplified sentence than for the original sentence. Using this method, we can also backtrack and seek out conversion errors by examining the cascade of partly simplified sentences.

4 Evaluation

To elucidate the effect of the sentence simplification, we applied the rules to five PPI corpora and evaluated the PPI extraction performance. We then analyzed the errors. The evaluation settings will be explained in Section 4.1. The results of the PPI extraction will be explained in Section 4.2. Finally, the deeper analysis results will be presented in Section 4.3.

4.1 Experimental Settings

The state-of-the-art PPI extraction system AkaneRE by Miwa et al. (2009) was used to evaluate our approach. The system uses a combination of three feature vectors: bag-of-words (BOW), shortest path (SP), and graph features. Classification models are trained with a support vector machine (SVM), and AkaneRE (with Mogura) is used with default parameter settings. The following two systems are used for a state-of-the-art comparison: AkaneRE with multiple parsers and corpora (Miwa et al., 2009), and Airola et al. (2008) single-parser, single-corpus system.

The rules were evaluated on the BioInfer (Pyysalo et al., 2007), AIMed (Bunescu et al., 2005), IEPA (Ding et al., 2002), HPRD50 (Fundel et al., 2006), and LLL (Nédellec, 2005) corpora[1]. Table 2 shows the number of positive (interacting) vs. all pairs. One duplicated abstract in the AIMed corpus was removed.

These corpora have several differences in their definition of entities and relations (Pyysalo et al., 2008). In fact, BioInfer and AIMed target all occurring entities related to the corpora (proteins, genes, etc). On the other hand, IEPA, HPRD50, and LLL only use limited named entities, based either on a list of entity names or on a named entity recognizer. Only BioInfer is annotated for other event types in addition to PPI, including static relations such as protein family membership. The sentence lengths are also different. The duplicated pair-containing sentences contain the following numbers of words on average: 35.8 in BioInfer, 31.3 in AIMed, 31.8 in IEPA, 26.5 in HPRD50, and 33.4 in LLL.

For BioInfer, AIMed, and IEPA, each corpus is split into training, development, and test datasets[2]. The training dataset from AIMed was the only training dataset used for validating the rules. The development datasets are used for error analysis. The evaluation was done on the test dataset, with models trained using training and development

[1] `http://mars.cs.utu.fi/PPICorpora/GraphKernel.html`

[2] This split method will be made public later.

	BioInfer		AIMed		IEPA		HPRD50		LLL	
	pos	all	pos	all	pos	all	pos	all	pos	all
training	1,848	7,108	684	4,072	256	630	-	-	-	-
development	256	928	102	608	23	51	-	-	-	-
test	425	1,618	194	1,095	56	136	-	-	-	-
all	2,534	9,653	980	5,775	335	817	163	433	164	330

Table 2: Number of positive (pos) vs. all possible sentence pairs in used PPI corpora.

	BioInfer			AIMed			IEPA		
Rule	Applied	F	AUC	Applied	F	AUC	Applied	F	AUC
No Application	0	62.5	83.0	0	61.2	87.9	0	73.4	82.5
Clause Selection	4,313	**63.5**	**83.9**	2,569	**62.5**	**88.2**	307	75.0	83.7
Entity Phrase	22,066	60.5	80.9	7,784	61.2	86.1	1,031	72.7	83.3
ALL	26,281	62.9	82.1	10,783	60.2	85.7	1,343	**75.4**	**85.7**

Table 3: Performance of PPI Extraction on test datasets. "Applied" represents the number of times the rules are applied on the corpus. "No Application" means PPI extraction without sentence simplification. ALL is the case all rules are used. The top scores for each corpus are shown in bold.

datasets). Ten-fold cross-validation (CV) was done to facilitate comparison with other existing systems. For HPRD50 and LLL, there are insufficient examples to split the data, so we use these corpora only for comparing the scores and statistics. We split the corpora for the CV, and measured the F-score (%) and area under the receiver operating characteristic (ROC) curve (AUC) as recommended in (Airola et al., 2008). We count each occurrence as one example because the correct interactions must be extracted for each occurrence if the same protein name occurs multiple times in a sentence.

In the experiments, the rules are applied in the following order: sentence–clause, exemplification, apposition, parentheses, coordination, copula, and relative-clause rules. Furthermore, if the same rule is applicable in different parts of the parse tree, then the rule is first applied closest to the leaf-nodes (deepest first). The order of the rules is arbitrary; changing it does not affect the results much. We conducted five experiments using the training and development dataset in IEPA, each time with a random shuffling of the order of the rules; the results were 77.8$\pm$0.26 in F-score and 85.9$\pm$0.55 in AUC.

4.2 Performance of PPI Extraction

The performance after rule application was better than the baseline (no application) on all the corpora, and most rules could be frequently applied. We show the PPI extraction performance on BioInfer, AIMed, and IEPA with rules of different groups in Table 3. The effect of using rules of different types for PPI extraction from HPRD50 and LLL is reported in Table 4 and Table 5. Table 6 shows the number of times each rule was applied in an "apply all-rules" experiment. The usability of the rules depends on the corpus, and different combinations of rules produce different

Rule	Applied	F	AUC
No Application	0	72.9	84.5
Sentence Clause	145	71.6	83.8
Relative Clause	7	73.3	84.1
Copula	0	72.9	84.5
Clause Selection	152	71.4	83.4
Apposition	64	73.2	84.6
Exemplification	33	72.9	84.7
Parentheses	90	72.9	85.1
Coordination	417	73.6	85.4
Entity Phrase	605	74.1	86.6
ALL	763	**75.0**	**86.6**

Table 4: Performance of PPI Extraction on HPRD50.

Rule	Applied	F	AUC
No Application	0	79.0	84.6
Sentence Clause	135	81.3	85.2
Relative Clause	42	78.8	84.6
Copula	0	79.0	84.6
Clause Selection	178	81.0	85.6
Apposition	197	79.6	83.9
Exemplification	0	79.0	84.6
Parentheses	56	79.5	85.8
Coordination	322	**84.2**	89.4
Entity Phrase	602	83.8	90.1
ALL	761	82.9	**90.5**

Table 5: Performance of PPI Extraction on LLL.

Rule	B	AIMed	IEPA	H	LLL
S. Cl.	3,960	2,346	300	150	135
R. Cl.	287	212	17	5	24
Copula	60	57	1	0	0
Cl. Sel.	4,307	2,615	318	155	159
Appos.	3,845	1,100	99	69	198
Exempl.	383	127	11	33	0
Paren.	2,721	2,158	235	91	88
Coord.	15,025	4,783	680	415	316
E. Foc.	21,974	8,168	1,025	608	602
Sum	26,281	10,783	1,343	763	761

Table 6: Distribution of the number of rules applied when all rules are applied. B:BioInfer, and H:HPRD50 corpora.

	Rules		Miwa et al.		Airola et al.	
	F	AUC	F	AUC	F	AUC
B	60.0	79.8	68.3	86.4	61.3	81.9
A	54.9	83.7	65.2	89.3	56.4	84.8
I	77.8	88.7	76.6	87.8	75.1	85.1
H	75.0	86.6	74.9	87.9	63.4	79.7
L	82.9	90.5	86.7	90.8	76.8	83.4

Table 7: Comparison with the results by Miwa et al. (2009) and Airola et al. (2008). The results with all rules are reported.

results. For the clause-selection rules, the performance was as good as or better than the baseline for all corpora, except for HPRD50, which indicates that the pair-containing clauses also include most of the important information for PPI extraction. Clause selection rules alone could improve the overall performance for the BioInfer and AIMed corpora. Entity-phrase rules greatly improved the performance on the IEPA, HPRD50, and LLL corpora, although these rules degraded the performance on the BioInfer and AIMed corpora. These phenomena hold even if we use small parts of the two corpora, so this is not because of the size of the corpora.

We compare our results with the results by Miwa et al. (2009) and Airola et al. (2008) in Table 7. On three of five corpora, our method provides better results than the state-of-the-art system by Airola et al. (2008), and also provides comparable results to those obtained using multiple parsers and corpora (Miwa et al., 2009) despite the fact that our method uses one parser and one corpus at a time. We cannot directly compare our result with Jonnalagadda and Gonzalez (2009) because the evaluation scheme, the baseline system, and test parts differ.

[FP→TN][Sentence, Parenthesis, Coordination] To characterize the AAV functions mediating this effect, cloned AAV type 2 wild-type or mutant genomes were transfected into simian virus 40 (SV40)-transformed hamster cells together with the six HSV replication genes (encoding UL5, UL8, major DNA-binding protein, **DNA polymerase**, UL42 , and **UL52**) which together are necessary and sufficient for the induction of SV40 DNA amplification (R. Heilbronn and H. zur Hausen, J. Virol. 63:3683-3692, 1989). (BioInfer.d760.s0)
[TP→FN][Coordination] Both the **GT155-calnexin** and the GT155-CAP-60 interactions were dependent on the presence of a correctly modified oligosaccharide group on GT155, a characteristic of many calnexin interactions. (AIMed.d167.s1408)
[TN→TN][Coordination, Parenthesis] **Leptin** may act as a negative feedback signal to the hypothalamic control of appetite through suppression of **neuropeptide Y** (NPY) secretion and stimulation of cocaine and amphetamine regulated transcript (CART) . (IEPA.d190.s454)

Figure 4: A rule-related error, a critical error, and a parser-related error. Regions removed by the rules are underlined, and target proteins are shown in bold. Predictions, applied rules, and sentence IDs are shown.

[FN→TP][Sentence, Coordination] **WASp** contains a binding motif for the Rho GTPase CDC42Hs as well as **verprolin** / cofilin-like actin-regulatory domains , but no specific actin structure regulated by CDC42Hs-WASp has been identified. (BioInfer.d795.s0)
[FN→TP][Parenthesis, Apposition] The protein **Raf-1** , a key mediator of mitogenesis and differentiation, associates with **p21ras** (refs 1-3) . (AIMed.d124.s1055)
[FN→TP][Sentence, Parenthesis] On the basis of far-Western blot and plasmon resonance (BIAcore) experiments, we show here that recombinant **bovine prion protein** (bPrP) (25-242) strongly interacts with the catalytic alpha/alpha' subunits of **protein kinase CK2** (also termed 'casein kinase 2') (IEPA.d197.s479)

Figure 5: Correctly simplified cases. The first sentence is a difficult (not PPI) relation, which is typed as "Similar" in the BioInfer corpus.

4.3 Analysis

We trained models using the training datasets and classified the examples in the development datasets. Two types of analysis were performed based on these results: *simplification-based* and *classification-based analysis*.

For the *simplification-based analysis*, we compared positive (interacting) and negative pair sentences that produce the exact same (inconsistent) sentence after protein names normalization and

	BioInfer				AIMed				IEPA				
Before simplification After simplification	FN TP	FP TN	TP FN	TN FP	FN TP	FP TN	TP FN	TN FP	FN TP	FP TN	TP FN	TN FP	Not Affected
No Error	18	2	3	35	14	21	21	8	3	2	0	4	32
No Application	3	2	0	3	0	7	8	0	0	1	0	1	7
Number of Errors	0	2	0	32	4	2	1	4	0	0	0	0	1
Number of Pairs	21	6	3	70	18	30	30	12	3	3	0	5	40
Coordination	0	0	0	20	4	2	1	0	0	0	0	0	1
Sentence	0	2	0	4	0	0	0	4	0	0	0	0	0
Parenthesis	0	0	0	5	0	0	0	0	0	0	0	0	0
Exemplification	0	0	0	2	0	0	0	0	0	0	0	0	0
Apposition	0	0	0	1	0	0	0	0	0	0	0	0	0

Table 8: Distribution of sentence simplification errors compared to unsimplified predictions with their types (on the three development datasets). TP, True Positive; TN, True Negative; FN, False Negative; FP, False Positive. "No Error" means that simplification was correct; "No Application" means that no rule could be applied; Other rule names mean that an error resulted from that rule application. "Not Affected" means that the prediction outcome did not change.

simplification in the training dataset. The numbers of such inconsistent sentences are 7 for BioInfer, 78 for AIMed, and 1 for IEPA. The few inconsistencies in BioInfer and IEPA are from errors by the rules, mainly triggered by parse errors. The frequent inconsistencies in AIMed are mostly from inconsistent annotations. For example, even if all coordinated proteins are either interacting or not, only the first protein mention is annotated as interacting.

For the *classification-based analysis*, we specifically examine simplified pairs that were predicted differently before and after the simplification. Pairs predicted differently before and after rule application were selected: 100 random pairs from BioInfer and all 90 pairs from AIMed. For IEPA, all 51 pairs are reported. Simplified results are classified as errors when the rules affect a region unrelated to the entities in the smallest sentence clause. The results of analysis are shown in Table 8. There were 34 errors in BioInfer, and 11 errors in AIMed. Among the errors, there were five *critical errors* (in two sentences, in AIMed). Critical errors mean that the pairs lost relation-related mentions, and the errors are the only errors which caused the changes in the truth-value of the relation. There was also a *rule-related error* (in BioInfer), which means that rules with correct parse results affect a region unrelated to the entities, and parse errors (*parser-related errors*). Figure 4 shows the rule-related error in BioInfer, one critical error in AIMed, and one parser-related error in IEPA.

5 Discussion

Our end goal is to provide consistent relation extraction for real tasks. Here we discuss the "safety" of applying our simplification rules, the difficulties in the BioInfer and AIMed corpora, the reduction of errors, and the requirements for such a general (PPI) extraction system.

Our rules are applicable to sentences, with little danger of changing the relation-related mentions. Figure 5 shows three successfully simplified cases ("No Error" cases from Table 8). The sentence simplification leaves sufficient information to determine the value of the relation in these examples. Relation-related mentions remained for most of the simplification error cases. There were only five critical errors, which changed the truth-value of the relation, out of 46 errors in 241 pairs shown in Table 8. Please note that some rules can be dangerous for other relation extraction tasks. For example, the *sentence clause rule* could remove modality information (negation, speculation, etc.) modifying the clause, but there are few such cases in the PPI corpora (see Table 8). Also, the task of hedge detection (Morante and Daelemans, 2009) can be solved separately, in the original sentences, after the interacting pairs have been found. For example, in the BioNLP shared task challenge and the BioInfer corpus, interaction detection and modality are treated as two different tasks. Once other NLP tasks, like static relation (Pyysalo et

al., 2009) or coreference resolution, become good enough, they can supplement or even substitute some of the proposed rules.

There are different difficulties in the BioInfer and AIMed corpora. BioInfer includes more complicated sentences and problems than the other corpora do, because 1) the apposition, coordination, and exemplification rules are more frequently used in the BioInfer corpus than in the other corpora (shown in Table 6), 2) there were more errors in the BioInfer corpus than in other corpora among the simplified sentences (shown in Table 8), and 3) BioInfer has more words per sentence and more relation types than the other corpora. AIMed contains several annotation inconsistencies as explained in Section 4.3. These inconsistencies must be removed to properly evaluate the effect of our method.

Simplification errors are mostly caused by parse errors. Our rule specifically examines a part of parser output; a probability is attached to the part. The probability is useful for defining the order of rule applications, and the n-best results by the parser are useful to fix major errors such as coordination errors. By using these modifications of rule applications and by continuous improvement in parsing technology for the biomedical domain, the performance on the BioInfer and AIMed corpora will be improved also for the all rules case.

The PPI extraction system lost the ability to capture some of the relation-related expressions left by the simplification rules. This indicates that the system used to extract some relations (before simplification) by using back-off features like bag-of-words. The system can reduce bad effects caused by parse errors, but it also captures the annotation inconsistencies in AIMed. Our simplification (without errors) can capture more general expressions needed for relation extraction. To provide consistent PPI relation extraction in a general setting (e.g. for multiple corpora or for other public text collections), the parse errors must be dealt with, and a relation extraction system that can capture (only) general relation-related expressions is needed.

6 Conclusion

We proposed a method to simplify sentences, particularly addressing the target entities for relation extraction. Using a few simple rules applicable to the output of a deep parser called Mogura, we showed that sentence simplification is effective for relation extraction. Applying all the rules improved the performance on three of the five corpora, while applying only the clause-selection rules raised the performance for the remaining two corpora as well. We analyzed the simplification results, and showed that the simple rules are applicable with little danger of changing the truth-values of the interactions.

The main contributions of this paper are: 1) explanation of general sentence simplification rules using HPSG for relation extraction, 2) presenting evidence that application of the rules improve relation extraction performance, and 3) presentation of an error analysis from two viewpoints: simplification and classification results.

As future work, we are planning to refine and complete the current set of rules, and to cover the shortcomings of the deep parser. Using these rules, we can then make better use of the parser's capabilities. We will also attempt to apply our simplification rules to other relation extraction problems than those of PPI.

Acknowledgments

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan), Genome Network Project (MEXT, Japan), and Scientific Research (C) (General) (MEXT, Japan).

References

Airola, Antti, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. A graph kernel for protein-protein interaction extraction. In *Proceedings of the BioNLP 2008 workshop*.

Bunescu, Razvan C. and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731.

Bunescu, Razvan C., 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.

Chun, Hong-Woo, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Naoki Nagata, Teruyoshi Hishiki, and Jun'ichi Tsujii. 2006. Extraction of gene-disease relations from medline using domain dictionaries and machine learning. In *The Pacific Symposium on Biocomputing (PSB)*, pages 4–15.

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

Doddington, George, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program: Tasks, data, and evaluation. In *Proceedings of LREC'04*, pages 837–840.

Dorr, Bonnie, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *in Proceedings of Workshop on Automatic Summarization*, pages 1–8.

Fundel, Katrin, Robert Küffner, and Ralf Zimmer. 2006. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Jonnalagadda, Siddhartha and Graciela Gonzalez. 2009. Sentence simplification aids protein-protein interaction extraction. In *Proceedings of the 3rd International Symposium on Languages in Biology and Medicine*, pages 109–114, November.

Matsuzaki, Takuya, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Efficient HPSG parsing with supertagging and cfg-filtering. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1671–1676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

Morante, Roser and Walter Daelemans. 2009. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the BioNLP 2009 Workshop*, pages 28–36, Boulder, Colorado, June. Association for Computational Linguistics.

Nédellec, Claire. 2005. Learning language in logic - genic interaction extraction challenge. In *Proceedings of the LLL'05 Workshop*.

Pyysalo, Sampo, 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*, 8:50.

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

Pyysalo, Sampo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. 2009. Static relations: a piece in the biomedical information extraction puzzle. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 1–9, Morristown, NJ, USA. Association for Computational Linguistics.

Sarawagi, Sunita. 2008. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377.

Vanderwende, Lucy, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Inf. Process. Manage.*, 43(6):1606–1618.

Vickrey, David and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio, June. Association for Computational Linguistics.

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