A Comparative Study of Syntactic Parsers for Event Extraction

Makoto Miwa[1] **Sampo Pyysalo**[1] **Tadayoshi Hara**[1] **Jun'ichi Tsujii**[1,2,3]
[1]Department of Computer Science, the University of Tokyo, Japan
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan.
[2]School of Computer Science, University of Manchester, UK
[3]National Center for Text Mining, UK
{mmiwa,smp,harasan,tsujii}@is.s.u-tokyo.ac.jp

Abstract

The extraction of bio-molecular events from text is an important task for a number of domain applications such as pathway construction. Several syntactic parsers have been used in Biomedical Natural Language Processing (BioNLP) applications, and the BioNLP 2009 Shared Task results suggest that incorporation of syntactic analysis is important to achieving state-of-the-art performance. Direct comparison of parsers is complicated by to differences in the such as the division between phrase structure- and dependency-based analyses and the variety of output formats, structures and representations applied. In this paper, we present a task-oriented comparison of five parsers, measuring their contribution to bio-molecular event extraction using a state-of-the-art event extraction system. The results show that the parsers with domain models using dependency formats provide very similar performance, and that an ensemble of different parsers in different formats can improve the event extraction system.

1 Introduction

Bio-molecular events are useful for modeling and understanding biological systems, and their automatic extraction from text is one of the key tasks in Biomedical Natural Language Processing (BioNLP). In the BioNLP 2009 Shared Task on event extraction, participants constructed event extraction systems using a variety of different parsers, and the results indicated that the use of a parser was correlated with high ranking in the task (Kim et al., 2009). By contrast, the results did not indicate a clear preference for a particular parser, and there has so far been no direct comparison of different parsers for event extraction.

While the outputs of parsers applying the same out format can be compared using a gold standard corpus, it is difficult to perform meaningful comparison of parsers applying different frameworks. Additionally, it is still an open question to what extent high performance on a gold standard treebank correlates with usefulness at practical tasks. Task-based comparisons of parsers provide not only a way to asses parsers across frameworks but also a necessary measure of their practical applicability.

In this paper, five different parsers are compared on the bio-molecular event extraction task defined in the BioNLP 2009 Shared Task using a state-of-the-art event extraction system. The data sets share abstracts with GENIA treebank, and the treebank is used as an evaluation standard. The outputs of the parsers are converted into two dependency formats with the help of existing conversion methods, and the outputs are compared in the two dependency formats. The evaluation results show that different syntactic parsers with domain models in the same dependency format achieve closely similar performance, and that an ensemble of different syntactic parsers in different formats can improve the performance of an event extraction system.

2 Bio-molecular Event Extraction with Several Syntactic Parsers

This paper focuses on the comparison of several syntactic parsers on a bio-molecular event extraction task with a state-of-the-art event extraction system. This section explains the details of the comparison. Section 2.1 presents the event ex-

Proceedings of the 2010 Workshop on Biomedical Natural Language Processing, ACL 2010, pages 37–45,
Uppsala, Sweden, 15 July 2010.

traction task setting, following that of the BioNLP 2009 Shared Task. Section 2.2 then summarizes the five syntactic parsers and three formats adopted for the comparison. Section 2.3 described how the state-of-the-art event extraction system of Miwa et al. (2010) is modified and used for the comparison.

2.1 Bio-molecular Event Extraction

The bio-molecular event extraction task considered in this study is that defined in the BioNLP 2009 Shared Task (Kim et al., 2009)[1]. The shared task provided common and consistent task definitions, data sets for training and evaluation, and evaluation criteria. The shared task consists of three subtasks: core event extraction (Task 1), augmenting events with secondary arguments (Task 2), and the recognition of speculation and negation of the events (Task 3) (Kim et al., 2009). In this paper we consider Task 1 and Task 2. The shared task defined nine event types, which can be divided into five simple events (Gene_expression, Transcription, Protein_catabolism, Phosphorylation, and Localization) that take one core argument, a multi-participant binding event (Binding), and three regulation events (Regulation, Positive_regulation, and Negative_regulation) that can take other events as arguments.

In the two tasks considered, events are represented with a textual trigger, type, and arguments, where the trigger is a span of text that states the event in text. In Task 1 the event arguments that need to be extracted are restricted to the core arguments Theme and Cause, and secondary arguments (locations and sites) need to be attached in Task 2.

2.2 Parsers and Formats

Five parsers and three formats are adopted for the evaluation. The parsers are GDep (Sagae and Tsujii, 2007)[2], the Bikel parser (Bikel) (Bikel, 2004)[3], the Charniak-Johnson reranking parser, using David McClosky's self-trained biomedical parsing model (MC) (McClosky, 2009)[4], the C&C CCG parser, adapted to biomedical text (C&C) (Rimell and Clark, 2009)[5], and the Enju parser with the GENIA model (Miyao et al., 2009)[6]. The formats are Stanford Dependencies (SD) (Figure 1), the CoNLL-X dependency format (Figure 2) and the predicate-argument structure (PAS) format used by Enju (Figure 3). With the exception of Enju, the analyses of these parsers were provided by the BioNLP 2009 Shared Task organizers. Analysis of system features in the task found that the use of parser output with one of

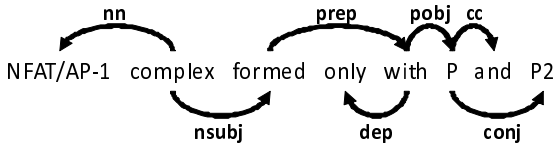

Figure 1: Stanford basic dependency tree

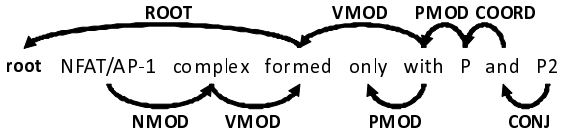

Figure 2: CoNLL-X dependency tree

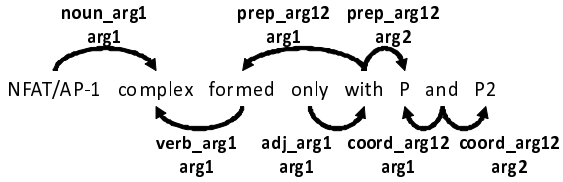

Figure 3: Predicate Argument Structure

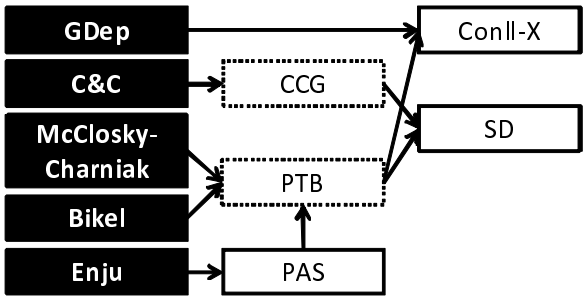

Figure 4: Format conversion dependencies in five parsers. Formats adopted for the evaluation is shown in solid boxes. SD: Stanford Dependency format, CCG: Combinatory Categorial Grammar output format, PTB: Penn Treebank format, and PAS: Predicate Argument Structure in Enju format.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/

[2] http://www.cs.cmu.edu/~sagae/parser/gdep/

[3] http://www.cis.upenn.edu/~dbikel/software.html

[4] http://www.cs.brown.edu/~dmcc/biomedical.html

[5] http://svn.ask.it.usyd.edu.au/trac/candc/

[6] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/

the formats considered here correlated with high rank at the task (Kim et al., 2009). A number of these parsers have also been shown to be effective for protein-protein interactions extraction (Miyao et al., 2009).

The five parsers operate in a number of different frameworks, reflected in their analyses. GDep is a native dependency parser that produces CoNLL-X-format dependency trees. MC and Bikel are phrase-structure parsers, and they produce Penn Treebank (PTB) format analyses. C&C is a deep parser based on Combinatory Categorial Grammar (CCG), and its native output is in a CCG-specific format. The output of C&C is converted into SD by a rule-based conversion script (Rimell and Clark, 2009). Enju is deep parser based on Head-driven Phrase Structure Grammar (HPSG) and produces a format containing predicate argument structures (PAS) along with a phrase structure tree in Enju format.

To study the contribution of the formats in which the five parsers output their analyses to task performance, we apply a number of conversions between the outputs, shown in Figure 4. The Enju PAS output is converted into Penn Treebank format using the method introduced by (Miyao et al., 2009). SD is generated from PTB by the Stanford tools (de Marneffe et al., 2006)[7], and CoNLL-X dependencies are generated from PTB by using Treebank Converter (Johansson and Nugues, 2007)[8]. We note that all of these conversions can introduce some errors in the conversion process.

With the exception of Bikel, all the applied parsers have models specifically adapted for biomedical text. Further, all of the biomedical domain models have been created with reference and for many parsers with direct training on the data of (a subset of) the GENIA treebank (Tateisi et al., 2005). The results of parsing with these models as provided for the BioNLP Shared Task are used in this comparison. However, we note that the shared task data, drawn from the GENIA event corpus (Kim et al., 2008), contains abstracts that are also in the GENIA treebank. This implies that the parsers are likely to perform better on the texts used in the shared task than on other biomedical domain text, and similarly that systems building on their output are expected to achieve best performance on this data. However, it does not invalidate comparison within the dataset. We further note that the models do not incorporate any knowledge of the event annotations of the shared task.

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

[8] http://nlp.cs.lth.se/software/treebank_converter/

2.3 Event Extraction System

The system by Miwa et al. (2010) is adopted for the evaluation. The system was originally developed for finding core events (Task 1 in the BioNLP 2009 Shared Task) using Enju and GDep with the native output of these parsers. The system consists of three supervised classification-based modules: a trigger detector, an event edge detector, and a complex event detector. The trigger detector classifies each word into the appropriate event types, the event edge detector classifies each edge between an event and a protein into an argument type, and the complex event detector classifies event candidates constructed by all edge combinations, deciding between event and non-event. The system uses one-vs-all support vector machines (SVMs) for the classifications.

The system operates on one sentence at a time, building features for classification based on the syntactic analyses for the sentence provided by the two parsers as well as the sequence of the words in the sentence, including the target candidate. The features include the constituents/words around entities (triggers and proteins), the dependencies, and the shortest paths among the entities. The feature generation is format-independent regarding the shared properties of different formats, but makes use also of format-specific information when available for extracting features, including the dependency tags, word-related information (e.g. a lexical entry in Enju format), and the constituents and their head information.

The previously introduced base system is here improved with two modifications. One modification is removing two classes of features from the original features (for details of the original feature representation, we refer to (Miwa et al., 2010)); specifically the features representing governor-dependent relationships from the target word, and the features representing each event edges in the complex event detector are removed. The other modification is to use head words in a trigger expression as a gold trigger word. This modification is inspired by the part-of-speech (POS) based selection proposed by Kilicoglu and Bergler (2009).

The system uses a head word "in" as a trigger word in a trigger expression "in the presence of" instead of using all the words of the expression. In cases where there is no head word information in a parser output, head words are selected heuristically: if a word does not modify another word in the trigger expression, the word is selected as a head word.

The system is also modified to find secondary arguments (Task 2 in the BioNLP 2009 Shared Task). The second arguments are treated as additional arguments in Task 1: the trigger detector finds secondary argument candidates, the event edge detector finds secondary argument edge candidates, and the complex event detector finds events including secondary arguments. The features are extracted using the same feature extraction method as for regulation events taking proteins as arguments.

3 Evaluation Setting

Event extraction performance is evaluated using the evaluation script provided by the BioNLP'09 shared task organizers[9] for the development data set, and the online evaluation system of the task[10] for the test data set. Results are reported under the official evaluation criterion of the task, i.e. the "Approximate Span Matching/Approximate Recursive Matching" criterion. Task 1 and Task 2 are solved at once for the evaluation.

As discussed in Section 2.2, the texts of the GENIA treebank are shared with the shared task data sets, which allows the gold annotations of the treebank to be used for reference. The GENIA treebank is converted into the Enju format with Enju. When the trees in the treebank cannot be converted into the Enju format, parse results are used instead. The GENIA treebank is also converted into PTB format[11]. The treebank is then converted into the dependency formats with the conversions described in Section 2.2. While based on manually annotated gold data, the converted treebanks are not always correct due to conversion errors.

The event extraction system described in Section 2.3 is used with the default settings shown in (Miwa et al., 2010). The positive and negative examples are balanced by placing more weight on the positive examples. The examples predicted with confidence greater than 0.5, as well as the examples with the most confident labels, are extracted. The C-values of SVMs are set to 1.0.

Some of the parse results do not include word base forms or part-of-speech (POS) tags, which are required by the event extraction system. To apply these parsers, the GENIA Tagger (Tsuruoka et al., 2005) output is adopted to add this information to the results.

[9] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/downloads.shtml

[10] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/eval-test.shtml

[11] http://categorizer.tmit.bme.hu/~illes/genia_ptb/

	BD	CD	CDP	CTD
Task 1	**55.60**	54.35	54.59	54.42
Task 2	**53.94**	52.65	52.88	52.76

Table 1: Comparison of the F-score results with different Stanford dependency variants on the development data set with the MC parser. Results for basic dependencies (BD), collapsed dependencies (CD), collapsed dependencies with propagation of conjunct dependencies (CDP), and collapsed tree dependencies (CTD) are shown. The best score in each task is shown in bold.

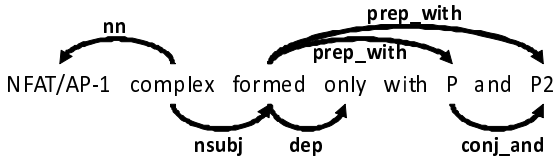

Figure 5: Stanford collapsed dependencies with propagation of conjunct dependencies

4 Evaluation

Results of event extraction with the setting in Section 2.3 will be presented in this section. Section 4.1 considers the effect of different variants of the Stanford Dependency representation. Section 4.2 presents the results of experiments with different parsers, and Section 4.3 shows the performance with ensembles of multiple parsers. Finally, the performance of the event extraction system is discussed in context of other proposed methods for the task in Section 4.4.

4.1 Stanford Dependency Setting

Stanford dependencies have four different variants: basic dependencies (BD), collapsed dependencies (CD), collapsed dependencies with propagation of conjunct dependencies (CDP), and collapsed tree dependencies (CTD) (de Marneffe and

	BD	CD	CDP	CTD
Task 1	54.22	**54.37**	53.88	53.84
	(-1.38)	(+0.02)	(-0.71)	(-0.58)
Task 2	52.73	**52.80**	52.31	52.35
	(-1.21)	(+0.15)	(-0.57)	(-0.41)

Table 2: Comparison of the F-score results with different Stanford dependency variants without dependency types.

Manning, 2008). Except for BD, these variants do not necessarily connect all the words in the sentence, and CD and CDP do not necessarily form a tree structure. Figure 5 shows an example of CDP converted from the tree in Figure 1. To select a suitable alternative for the comparative experiments, we first compared these variants as a preliminary experiment. Table 1 shows the comparison results with the MC parser. Dependencies are generalized by removing expressions after "_" of the dependencies (e.g. "_with" in prep_with) for better performance. We find that basic dependencies give the best performance to event extraction, with little difference between the other variants. This result is surprising, as variants other than basic have features such as the resolution of conjunctions that are specifically designed for practical applications. However, basic dependendencies were found to consistently provide best performance also for the other parsers[12].

The SD variants differ from each other in two key aspects: the dependency structure and the dependency types. To gain insight into why the basic dependencies should provide better performance than other variants, we performed an experiment attempting to isolate these factors by repeating the evaluation while eliminating the dependency types. The results of this evaluation are shown in Table 2. The results indicate that the contribution of the dependency types to extraction performance differs between the variants: the expected performance drop is most notable for the basic dependencies, and for the collapsed dependencies there is even a minute increase in performance, making results for collapsed dependencies best of the untyped results (by a very narrow margin). While this result doesn't unambiguously point to a specific explanation for why basic dependencies provide best performance when types are not removed, possible explanations include errors in typing or sparseness issues causing problems in generalization for the types of non-basic dependencies. While achieving a clear resolution of the results of the comparison between SD variants requires more analysis, from a performance optimization perspective the results present an uncomplicated choice. Thus, in the following evaluation, the basic dependencies are adopted for all SD results.

[12]Collapsed tree dependencies are not evaluated on the C&C parser since the conversion is not provided.

4.2 Parser Comparison

Results with different parsers and different formats on the development data set are summarized in Table 3. Baseline results are produced by removing dependency (or PAS) information from the parse results. The baseline results differ between the represetations as the word base forms and POS tags produced by the GENIA tagger for use with the Stanford dependency and CoNLL-X formats are different from those for Enju, and because head word information in Enju format is used. The evaluation finds best results for both tasks with Enju, using its native output format. However, as discussed in Section 2.3, the treatment of the Enju format and the other two formats are slightly different, this result does not necessarily indicate that the Enju format is the best alternative for event extraction.

Unsurprisingly, we find that the Bikel parser, the only one in the comparison lacking a model adapted to the biomedical domain, performs worse than the other parsers. For SD, we find best results for C&C, which is notable as the parser output is processed into SD by a custom conversion, while MC output uses the *de facto* conversion of the Stanford tools. Similarly, MC produces the best result for the CoNLL-X format, which is the native output format of GDep. Enju and GDep produces comparable results to the best formats for both tasks. Overall, we find that event extraction results for the parsers applying GENIA treebank models are largely comparable for the dependency formats (SD and CoNLL-X).

The results with the data derived from the GENIA treebank can be considered as upper bounds for the parsers and formats at the task, although conversion errors are expected to lower these bounds to some extent. Even though trained on the treebank, using the parsers does not provide performance as high as that for using the GE-

	Task 1			Task 2		
	SD	CoNLL	PAS	SD	CoNLL	PAS
Baseline	51.05	-	50.42	49.17	-	48.88
GDep	-	55.70	-	-	54.37	-
Bikel	53.29	53.22	-	51.40	51.27	-
MC	55.60	56.01	-	53.94	54.51	-
C&C	56.09	-	-	54.27	-	-
Enju	55.48	55.74	**56.57**	54.06	54.37	**55.31**
GENIA	56.34	56.09	57.94	55.04	54.57	56.40

Table 3: Comparison of F-score results with five parsers in three different formats on the development data set. SD: Stanford basic Dependency format, CoNLL: CoNLL-X format, and PAS: Predicate Argument Structure in Enju format. Results without dependency (or PAS) information are shown as baselines. The results with the GENIA treebank (converted into PTB format and Enju format) are shown for comparison (GENIA). The best score in each task is shown in bold, and the best score in each task and format is underlined.

	Task 1			Task 2		
	C&C SD	MC CoNLL	Enju CoNLL	C&C SD	MC CoNLL	Enju CoNLL
MC CoNLL	57.44 (+1.35)	- -	- -	55.75 (+1.24)	- -	- -
Enju CoNLL	56.47 (+0.38)	56.24 (+0.23)	- -	54.85 (+0.48)	54.70 (+0.19)	- -
Enju PAS	57.20 (+0.63)	**57.78** (+1.21)	56.59 (+0.02)	55.75 (+0.44)	**56.39** (+1.08)	55.12 (-0.19)

Table 4: Comparison of the F-score results with parser ensembles on the development data set. C&C with Stanford basic Dependency format, MC with CoNLL-X format, Enju with CoNLL-X format, and Enju with Predicate Argument Structure in Enju format are used for the parser ensemble. The changes from single-parser results are shown in parentheses. The best score in each task is shown in bold.

NIA treebank, but in many cases results with the parsers are only slightly worse than results with the treebank. The results suggest that there is relative little remaining benefit to be gained for event extraction from improving parser performance. This supports the claim that most of the errors in event extraction are not caused by the parse errors in (Miwa et al., 2010). Experiments using the CoNLL-X format produce slightly worse results than for SD with the gold treebank data, which is at variance with the indication from parser-based results with MC and Enju. Thus, the results do not provide any systematic indication suggesting that one dependency format would be superior to the other in use for event extraction.

4.3 Event Extraction with Parser Ensemble

The four parser outputs were selected for the evaluation of a parser ensemble: C&C with Stanford basic Dependency format, MC with CoNLL-X format, Enju with CoNLL-X format, and Enju with Predicate Argument Structure in Enju format. Table 4 summarizes the parser ensemble results. We find that all ensembles of different parsers in different formats produce better results than those for single parser outputs (Table 3); by contrast, the results indicate that ensembles of the same formats (MC + Enju in CoNLL-X format) or parsers (Enju in CoNLL-X and Enju formats) produce relatively small improvements, may in some cases even reduce performance. The results thus indicate that while a parser ensemble can be effective but that it is important to apply different parsers in different formats.

Table 5 shows detailed results with three parsers with three different formats. The ensembles systematically improve F-scores in regulation and the overall performance ("All"), but the ensembles can degrade the performance for simple and binding events. Different parser outputs are shown to have their strengths and weaknesses in different event groups. The use of Enju, for exam-

	Simple	Binding	Regulation	All
		Task 1		
BL-E	75.85 / 71.09 / 73.39	40.32 / 38.17 / 39.22	30.65 / 48.16 / 37.46	46.12 / 55.60 / 50.42
BL-G	76.03 / 73.48 / 74.73	40.32 / 38.17 / 39.22	33.50 / 45.95 / 38.75	47.74 / 54.86 / 51.05
C	78.89 / 78.43 / 78.66	48.79 / 43.37 / 45.92	37.17 / 54.07 / 44.06	51.82 / 61.12 / 56.09
M	79.79 / 77.12 / 78.43	43.95 / 41.13 / 42.50	39.41 / 52.94 / 45.18	52.66 / 59.82 / 56.01
E	79.79 / 76.07 / 77.88	45.16 / 43.75 / 44.44	40.12 / 53.68 / 45.92	53.21 / 60.38 / 56.57
C+M	**80.50** / **79.05** / **79.77**	48.39 / 42.25 / 45.11	41.85 / 53.17 / 46.84	54.84 / 60.31 / 57.44
C+E	79.79 / 76.46 / 78.09	47.98 / **45.59** / 46.76	41.04 / 53.66 / 46.51	54.11 / 60.66 / 57.20
E+M	**80.50** / 77.15 / 78.79	44.35 / 42.97 / 43.65	42.26 / **55.63** / **48.03**	54.50 / **61.49** / 57.78
C+E+M	80.14 / 77.07 / 78.58	**51.61** / 42.95 / **46.89**	**42.46** / 54.30 / 47.66	**55.51** / 60.27 / **57.79**
		Task 2		
BL-E	74.60 / 69.10 / 71.75	36.55 / 34.73 / 35.62	29.89 / 47.20 / 36.60	44.74 / 53.86 / 48.88
BL-G	74.42 / 71.31 / 72.83	36.55 / 33.33 / 34.87	32.52 / 44.83 / 37.70	46.13 / 52.64 / 49.17
C	77.64 / 76.77 / 77.20	43.78 / 38.79 / 41.13	36.17 / 52.89 / 42.96	50.14 / 59.14 / 54.27
M	78.71 / 75.95 / 77.31	39.36 / 36.57 / 37.91	38.70 / 52.12 / 44.42	51.25 / 58.21 / 54.51
E	79.07 / 75.26 / 77.12	41.37 / 40.08 / 40.71	39.31 / 52.86 / 45.09	51.98 / 59.10 / 55.31
C+M	79.61 / **78.03** / **78.81**	43.37 / 36.99 / 39.93	40.93 / 52.07 / 45.83	53.31 / 58.41 / 55.75
C+E	78.89 / 75.34 / 77.08	44.18 / **40.89** / **42.47**	40.22 / 52.86 / 45.68	52.81 / 59.04 / 55.75
E+M	**79.79** / 76.33 / 78.02	40.16 / 38.76 / 39.45	41.34 / **54.69** / **47.09**	53.15 / **60.05** / **56.39**
C+E+M	79.43 / 76.25 / 77.81	**46.18** / 37.46 / 41.37	**41.54** / 53.39 / 46.72	**53.98** / 58.45 / 56.13

Table 5: Comparison of Recall / Precision / F-score results on the development data set. C&C with Stanford basic Dependency format (C), MC with CoNLL-X format (M), and Enju with Predicate Argument Structure in Enju format (E) are used for the evaluation. Results with Enju output without PAS information (BL-E) and the GENIA tagger output (BL-G) are shown as baselines. Results on simple, binding, regulation, and all events are shown. The best score in each result is shown in bold.

	Simple	Binding	Regulation	All
		Task 1		
Ours	**67.09** / **77.59** / **71.96**	49.57 / 51.65 / 50.59	**38.42** / **53.95** / **44.88**	**50.28** / **63.19** / **56.00**
Miwa	65.31 / 76.44 / 70.44	**52.16** / **53.08** / **52.62**	35.93 / 46.66 / 40.60	48.62 / 58.96 / 53.29
Björne	64.21 / 77.45 / 70.21	40.06 / 49.82 / 44.41	35.63 / 45.87 / 40.11	46.73 / 58.48 / 51.95
Riedel	N/A	23.05 / 48.19 / 31.19	26.32 / 41.81 / 32.30	36.90 / 55.59 / 44.35
		Task 2		
Ours	**65.77** / **75.29** / **70.21**	**47.56** / **49.55** / **48.54**	**38.24** / **53.57** / **44.62**	**49.48** / **61.87** / **54.99**
Riedel	N/A	22.35 / 46.99 / 30.29	25.75 / 40.75 / 31.56	35.86 / 54.08 / 43.12

Table 6: Comparison of Recall / Precision / F-score results on the test data set. MC with CoNLL-X format and Enju with Predicate Argument Structure in Enju format are used for the evaluation. Results on simple, binding, regulation, and all events are shown. Results by Miwa et al. (2010) (Miwa), Björne et al. (2009) (Björne), and Riedel et al. (2009) (Riedel) for Task 1 and Task 2 are shown for comparison. The best score in each result is shown in bold.

ple, is good for extracting regulation events, but produced weaker results for simple events. The ensembles of two parser outputs inherit both the strengths and weaknesses of the outputs in most cases, and the strengths and weaknesses of the ensembles vary depending on the combined parser outputs. The differences in performance between ensembles of the outputs of two parsers to the ensemble of the three parser outputs are +0.01 for Task 1, and -0.26 for Task 2. This result suggests that adding more different parsers does not always improve the performance. The ensemble of three parser outputs, however, shows stable performance across categories, scoring in the top two for binding, regulation, and all events, in the top four for simple events.

4.4 Performance of Event Extraction System

Table 6 shows a comparison of performance on the shared task test data. MC with CoNLL-X format and Enju with Predicate Argument Structure in Enju format are used for the evaluation, selecting one of the best performing ensemble settings in Section 4.3. The performance of the best systems in the original shared task is shown for reference ((Björne et al., 2009) in Task 1 and (Riedel et al., 2009) in Task 2). The event extraction system with our modifications performed significantly better than the best systems in the shared task, further outperforming the original system by Miwa et al. (2010). This result shows that the system applied for the comparison of syntactic parsers achieves state-of-the-art performance at event extraction. This result also shows that the system originally developed only for core events extraction can be easily extended for other arguments simply by treating the other arguments as additional arguments.

5 Related Work

Many approaches for parser comparison have been proposed in the BioNLP field. Most comparisons have used gold treebanks with intermediate formats (Clegg and Shepherd, 2007; Pyysalo et al., 2007). Application-oriented parser comparison across several formats was first introduced by Miyao et al. (2009), who compared eight parsers and five formats for the protein-protein interaction (PPI) extraction task. PPI extraction, the recognition of binary relations of between proteins, is one of the most basic information extraction tasks in the BioNLP field. Our findings do not conflict with those of Miyao et al. Event extraction can be viewed as an additional extrinsic evaluation task for syntactic parsers, providing more reliable and evaluation and a broader perspective into parser performance. An additional advantage of application-oriented evaluation on BioNLP shared task data is the availability of a manually annotated gold standard treebank, the GENIA treebank, that covers the same set of abstracts as the task data. This allows the gold treebank to be considered as an evaluation standard, in addition to comparison of performance in the primary task.

6 Conclusion

We compared five parsers and three formats on a bio-molecular event extraction task with a state-of-the-art event extraction system. The specific task considered was the BioNLP shared task, allowing the use of the GENIA treebank as a gold standard parse reference. The event extraction system, modified for a higher performance and an additional subtask, showed high performance on the shared task subtasks considered. Four of the five considered parsers were applied using biomedical models trained on the GENIA treebank, and they were found to produce similar performance. Parser ensembles were further shown to allow improvement of the performance of the event extraction system.

The contributions of this paper are 1) the comparison of several commonly used parsers on the event extraction task with a gold treebank, 2) demonstration of the usefulness of the parser ensemble on the task, and 3) the introduction of a state-of-the-art event extraction system. One limitation of this study is that the comparison between the parsers is not perfect, as the format conversions miss some information from the original formats and results with different formats depend on the ability of the event extraction system to take advantage of their strengths. To maximize comparability, the system was designed to extract features identically from similar parts of the dependency-based formats, further adding information provided by other formats, such as the lexical entries of the Enju format, from external resources. The results of this paper are expected to be useful as a guide not only for parser selection for biomedical information extraction but also for the development of event extraction systems.

The selection of compared parsers and formats in the present evaluation is somewhat limited. As future work, it would be informative to extend the comparison to other syntactic representations, such as the PTB format. Finally, the evaluation showed that the system fails to recover approximately 40% of events even when provided with manually annotated treebank data, showing that other methods and resources need to be adopted to further improve bio-molecular event extraction systems. Such improvement is left as future work.

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

Daniel M. Bikel. 2004. A distributional analysis of a lexicalized statistical parsing model. In *In EMNLP*, pages 182–189.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the BioNLP'09 Shared Task on Event Extraction*, pages 10–18.

Andrew B. Clegg and Adrian J. Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. Technical report, September.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*, Tartu, Estonia, May 25-26.

Halil Kilicoglu and Sabine Bergler. 2009. Syntactic dependency based heuristics for biological event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 119–127, Boulder, Colorado, June. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9:10.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 1–9.

David McClosky. 2009. *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. Ph.D. thesis, Department of Computer Science, Brown University.

Makoto Miwa, Rune Sætre, Jin-Dong Kim, and Jun'ichi Tsujii. 2010. Event extraction with complex event classification using rich features. *Journal of Bioinformatics and Computational Biology (JBCB)*, 8(1):131–146, February.

Yusuke Miyao, Kenji Sagae, Rune Sætre, Takuya Matsuzaki, and Jun ichi Tsujii. 2009. Evaluating contributions of natural language parsers to protein-protein interaction extraction. *Bioinformatics*, 25(3):394–400.

Sampo Pyysalo, Filip Ginter, Veronika Laippala, Katri Haverinen, Juho Heimonen, and Tapio Salakoski. 2007. On the unification of syntactic annotations under the stanford dependency scheme: A case study on bioinfer and genia. In *Biological, translational, and clinical language processing*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A markov logic approach to bio-molecular event extraction. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 41–49, Morristown, NJ, USA. Association for Computational Linguistics.

Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *J. of Biomedical Informatics*, 42(5):852–865.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *EMNLP-CoNLL 2007*.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Junfichi Tsujii. 2005. Syntax annotation for the genia corpus. In *Proceedings of the IJCNLP 2005, Companion volume*, pages 222–227, Jeju Island, Korea, October.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In Panayiotis Bozanis and Elias N. Houstis, editors, *Panhellenic Conference on Informatics*, volume 3746 of *Lecture Notes in Computer Science*, pages 382–392. Springer.