Evaluating Dependency Representation 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
[2]School of Computer Science, University of Manchester
[3]National Center for Text Mining
{mmiwa,smp,harasan,tsujii}@is.s.u-tokyo.ac.jp

Abstract

The detailed analyses of sentence structure provided by parsers have been applied to address several information extraction tasks. In a recent bio-molecular event extraction task, state-of-the-art performance was achieved by systems building specifically on dependency representations of parser output. While intrinsic evaluations have shown significant advances in both general and domain-specific parsing, the question of how these translate into practical advantage is seldom considered. In this paper, we analyze how event extraction performance is affected by parser and dependency representation, further considering the relation between intrinsic evaluation and performance at the extraction task. We find that good intrinsic evaluation results do not always imply good extraction performance, and that the types and structures of different dependency representations have specific advantages and disadvantages for the event extraction task.

1 Introduction

Advanced syntactic parsing methods have been shown to effective for many information extraction tasks. The BioNLP 2009 Shared Task, a recent bio-molecular event extraction task, is one such task: analysis showed that the application of a parser correlated with high rank in the task (Kim et al., 2009). The automatic extraction of biomolecular events from text is important for a number of advanced domain applications such as pathway construction, and event extraction thus a key task in Biomedical Natural Language Processing (BioNLP).

Methods building feature representations and extraction rules around dependency representations of sentence syntax have been successfully applied to a number of tasks in BioNLP. Several parsers and representations have been applied in high-performing methods both in domain studies in general and in the BioNLP'09 shared task in particular, but no direct comparison of parsers or representations has been performed. Likewise, a number of evaluation of parser outputs against gold standard corpora have been performed in the domain, but the broader implications of the results of such intrinsic evaluations are rarely considered. The BioNLP'09 shared task involved documents contained also in the GENIA treebank (Tateisi et al., 2005), creating an opportunity for direct study of intrinsic and task-oriented evaluation results. As the treebank can be converted into various dependency formats using existing format conversion methods, evaluation can further be extended to cover the effects of different representations.

In this this paper, we consider three types of dependency representation and six parsers, evaluating their performance from two different aspects: dependency-based intrinsic evaluation, and effectiveness for bio-molecular event extraction with a state-of-the-art event extraction system. Comparison of intrinsic and task-oriented evaluation re-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 779–787,
Beijing, August 2010

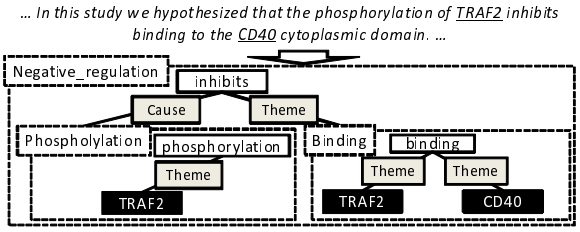


Figure 1: Event Extraction.

sults shows that performance against gold standard annotations is not always correlated with event extraction performance. We further find that the dependency types and overall structures employed by the different dependency representations have specific advantages and disadvantages for the event extraction task.

2 Bio-molecular Event Extraction

In this study, we adopt the event extraction task defined in the BioNLP 2009 Shared Task (Kim et al., 2009) as a model information extraction task. Figure 1 shows an example illustrating the task of event extraction from a sentence. The shared task provided common and consistent task definitions, data sets for training and evaluation, and evaluation criteria. The shared task defined 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) used to capture both biological regulation and general causation. The participants of simple and Binding events were specified to be of the general Protein type, while regulation-type events could also take other events as arguments, creating complex event structures.

We consider two subtasks, Task 1 and Task 2, out of the three defined in the shared task. Task 1 focuses on core event extraction, and Task 2 involves augmenting extracted events with secondary arguments (Kim et al., 2009). 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 Theme and Cause roles, with secondary arguments corresponding to locations and sites considered in Task 2.

2.1 Event Extraction System

For evaluation, we apply the system of Miwa et al. (2010b). The system was originally developed for finding core events (Task 1) using the native output of the Enju and GDep 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 candidate participant 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 classification.

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.

We apply here a variant of the base system incorporating a number of modifications. The applied system performs feature selection removing two classes of features that were found not to be beneficial for extraction performance, and applies a refinement of the trigger expressions of events. The system is further extended to find also secondary arguments (Task 2). For a detailed description of these improvements, we refer to Miwa et al. (2010a).

3 Parsers and Representations

Six publicly available parsers and three dependency formats are considered in this paper. The

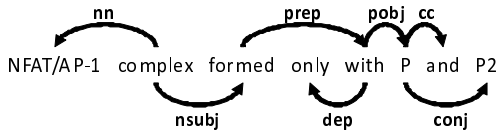


Figure 2: Stanford basic dependency tree

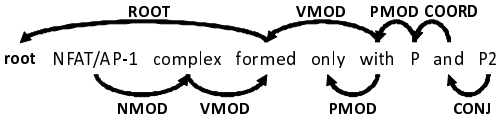


Figure 3: CoNLL-X dependency tree

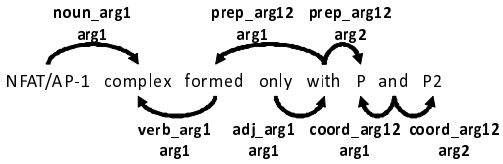


Figure 4: Predicate Argument Structure

parsers are GDep (Sagae and Tsujii, 2007), the Bikel parser (Bikel) (Bikel, 2004), the Stanford parser with two probabilistic context-free grammar (PCFG) models[1] (Wall Street Journal (WSJ) model (Stanford WSJ) and "augmented English" model (Stanford eng)) (Klein and Manning, 2003), the Charniak-Johnson reranking parser, using David McClosky's self-trained biomedical parsing model (MC) (McClosky, 2009), the C&C CCG parser, adapted to biomedical text (C&C) (Rimell and Clark, 2009), and the Enju parser with the GENIA model (Miyao et al., 2009). The formats are Stanford Dependencies (SD) (Figure 2), the CoNLL-X dependency format (CoNLL) (Figure 3) and the predicate-argument structure (PAS) format used by Enju (Figure 4). With the exception of Stanford and Enju, the analyses of these parsers were provided by the BioNLP 2009 Shared Task organizers.

The six parsers operate in a number of different frameworks, reflected in their analyses. GDep is a native dependency parser that produces CoNLL dependency trees, with dependency types similar to those of CoNLL 2007. Bikel, Stanford, and MC

[1]Experiments showed no benefit from using the lexicalized models with the Stanford parser.

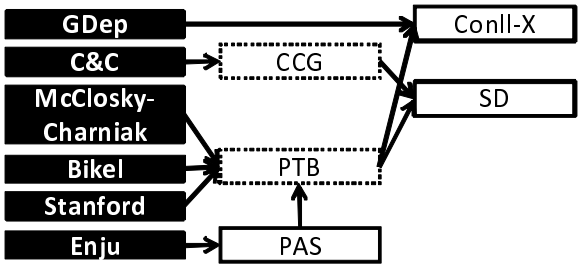


Figure 5: Format conversion dependencies in six parsers. Formats adopted for the evaluation are shown in solid boxes. SD: Stanford Dependency format, CCG: Combinatory Categorial Grammar output format, PTB: Penn Treebank format, and PAS: Predicate Argument Structure in Enju format.

are phrase-structure parsers trained on Penn Treebank format (PTB) style treebanks, and they produce PTB trees. 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 can be 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 along with a phrase structure tree in Enju format, which can be converted into PTB format (Miyao et al., 2009).

For direct comparison and for the study of contribution of the formats in which the six parsers output their analyses to task performance, we apply a number of conversions between the outputs, shown in Figure 5. The Enju PAS output is converted into PTB using the method introduced by (Miyao et al., 2009). SD is generated from PTB by the Stanford tools (de Marneffe et al., 2006), and CoNLL generated from PTB by using Treebank Converter (Johansson and Nugues, 2007). With the exception of GDep, all CoNLL outputs are generated by the conversion and thus share dependency types. We note that all of these conversions can introduce some errors in the conversion process.

4 Evaluation Setting

4.1 Event Extraction Evaluation

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

The event extraction system described in Section 2.1 is used with the default settings given in (Miwa et al., 2010b). The C-values of SVMs are set to 1.0, but 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. Task 1 and Task 2 are solved at once for the evaluation.

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.

4.2 Dependency Representation Evaluation

The parsers are evaluated with precision, recall, and F-score for each dependency type. We note that the parsers may produce more fine-grained word segmentations than that of the gold standard: for example, two words "p70(S6)-kinase activation" in the gold standard tree (Figure 6 (a)) is segmented into five words by Enju (Figure 6 (b)). In the evaluation the word segmentations in the gold tree are used, and dependency transfer and word-based normalization are performed to match parser outputs to these. Dependencies related to the segmentations are transferred to the enclosing word as follows. If one word is segmented into several segments by a parser, all the dependencies between the segments are removed (Figure 6 (c)) and the dependency between another word and the segments is converted into the dependency between the two words (Figure 6 (d)).

The parser outputs in SD and CoNLL can be assumed to be trees, so each node in the tree have only one parent node. However, in the converted tree nodes can have more than one parent. We cannot simply apply accuracy, or (un)labeled attachment score[3]. Word-based normalization is performed to avoid negative impact by the word segmentations by parsers. When (a) and (d) in Figure 6 are compared, the counts of correct relations will be 1.0 (0.5 for upper NMOD and 0.5 for lower NMOD in Figure 6 (d)) for the parser (precision), and the counts of correct relations will be 1.0 (for NMOD in Figure 6 (a)) for the gold (recall). This F-score is a good approximation of accuracy.

4.3 GENIA treebank processing

For comparison and evaluation, the texts in the GENIA treebank (Tateisi et al., 2005) are converted to the various formats as follows. To create PAS, the treebank is converted with Enju, and for trees that fail conversion, parse results are used instead. The GENIA treebank is also converted into PTB[4], and then converted into SD and CoNLL as described in Section 3. While based on manually annotated gold data, the converted treebanks are not always correct due to conversion errors.

5 Evaluation

This section presents evaluation results. Intrinsic evaluation is first performed in Section 5.1. Section 5.2 considers the effect of different SD variants. Section 5.3 presents the results of experiments with different parsers. Section 5.4 shows the performance of different parsers. Finally, the performance of the event extraction system is discussed in context of other proposed methods for the task in Section 5.5.

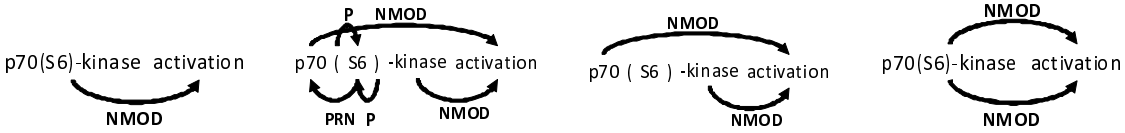
5.1 Intrinsic Evaluation

We initially briefly consider the results of an intrinsic evaluation comparing parser outputs to reference data automatically derived from the gold standard treebank. Table 1 shows results for the parsers whose outputs could be converted into the

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

[3] http://nextens.uvt.nl/~conll/

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



(a) Gold Word Segmentations

(b) Parser Word Segmentations

(c) Inner Dependency Removal

(d) Dependency Transfer

Figure 6: Example of Word Segmentations of the words by gold and Enju and Dependency Transfer.

	Typed						Untyped					
	SD			CoNLL			SD			CoNLL		
	P	R	F	P	R	F	P	R	F	P	R	F
Bikel	70.31	70.37	70.34	77.81	77.56	77.69	80.54	80.60	80.57	82.43	82.18	82.31
SP WSJ	74.11	73.94	74.03	81.41	81.47	81.44	81.36	81.16	81.26	84.05	84.05	84.05
SP eng	79.08	78.89	78.98	84.92	84.82	84.87	84.16	83.96	84.06	86.54	86.47	86.51
C&C	80.31	78.04	79.16		-		84.91	82.28	83.57		-	
MC	79.56	79.63	79.60	88.13	87.87	88.00	87.43	87.50	87.47	89.81	89.42	89.62
Enju	85.59	85.62	85.60	88.59	89.51	89.05	88.28	88.30	88.29	90.24	90.77	90.50

Table 1: Comparison of precision, recall, and F-score results with five parsers (two models for Stanford) in two different formats on the development data set (SP abbreviates for Stanford Parser). Results shown separately for evaluation including dependency types and one eliminating them. Parser/model combinations above the line do not use in-domain data, others do.

SD and CoNLL dependency representations using the Stanford tools and Treebank Converter, respectively. For Stanford, both the Penn Treebank WSJ section and "augmented English" (eng) models were tested; the latter includes biomedical domain data. The Enju results for PAS are 91.48 with types and 93.39 without in F-score. GDep not shown as its output is not compatible with that of Treebank Converter.

Despite numerical differences, the two representations and two criteria (typed/untyped) all produce largely the same ranking of the parsers.[5] The evaluations also largely agree on the magnitude of the reduction in error afforded through the use of in-domain training data for the Stanford parser, with all estimates falling in the 15-19% range. Similarly, all show substantial differences between the parsers, indicating e.g. that the error rate of Enju is 50% or less of that of Bikel.

These results serve as a reference point for extrinsic evaluation results. However, it should be noted that as the parsers make use of annotated domain training data to different extents, this evaluation does not provide a sound basis for direct comparison of the parsers themselves.

[5]One larger divergence is between typed and untyped SD results for MC. Analysis suggest one cause is frequent errors in tagging hyphenated noun-modifiers such as *NF-kappaB* as adjectives.

	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 2: Comparison of the F-score results with different SD variants on the development data set with the MC parser. The best score in each task is shown in bold.

5.2 Stanford Dependency Setting

SD 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 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. Table 2 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 dependencies were found to consistently provide best performance also for the other parsers[6]. Thus, in the following evaluation, the basic dependencies are adopted for all SD results.

5.3 Parser Comparison on Event Extraction

Results with different parsers and different formats on the development data set are summarized in Table 3. Baseline results are produced by removing dependency information from the parse results. The baseline results differ between the representations as the word base forms and POS tags produced by the GENIA tagger for use with SD and CoNLL are different from PAS, and because head word information in the Enju format is used. The evaluation finds best results for both tasks with Enju, using its native output format. However, as discussed in Section 2.1, the treatment of PAS and the other two formats are slightly different, this result does not necessarily indicate that PAS is the best alternative for event extraction.

The Bikel and Stanford WSJ parsers, lacking models adapted to the biomedical domain, performs mostly worse than the other parsers. The other parsers, even though trained on the treebank, do not provide performance as high as that for using the GENIA treebank, but, with the exception of Stanford eng with CoNLL, results with the parsers are only slightly worse than results with the treebank. 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. The results suggest that there is relative little remaining benefit to be gained from improving parser performance.

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

5.4 Effects of Dependency Representation

Intrinsic evaluation results (Section 5.1) cannot be used directly for comparing the parsers, since some of the parsers contain models trained on the GENIA treebank. To investigate the effects of the evaluation results to the event extraction, we performed event extraction with eliminating the dependency types. Table 4 summarizes the results with the dependency structures (without the dependency types) on the development data set. Interestingly, we find the performance increases in Bikel and Stanford by eliminating the dependency types. This implies that the inaccurate dependency types shown in Table 1 confused the event extraction system. SD and PAS drops more than CoNLL, and Enju with CoNLL structures perform best in total when the dependency types are removed. This result shows that the formats have their own strengths in finding events, and CoNLL structure with SD or PAS types can be a good representation for the event extraction.

By comparing Table 3, Table 1, and Table 4, we found that the better dependency performance does not always produce better event extraction performance especially when the difference of the dependency performance is small. MC and Enju results show that performance in dependency is important for event extraction. SD can be better than CoNLL for the event extraction (shown with the gold treebank data in Table 3), but the types and relations of CoNLL were well predicted, and MC and Enju performed better for CoNLL than for SD in total.

5.5 Performance of Event Extraction System

Several systems are compared by the extraction performance on the shared task test data in Table 5. GDep and Enju with PAS are used for the evaluation, which is the same evaluation setting with the original system by Miwa et al. (2010b). 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 performs significantly better than the best systems in the shared task, further outperforming the original system. This shows that the comparison of the parsers is performed with a state-of-the-art sys-

	Task 1			Task 2		
	SD	CoNLL	PAS	SD	CoNLL	PAS
Baseline	51.05	-	50.42	49.17	-	48.88
Bikel	53.29	53.22	-	51.40	51.27	-
Stanford WSJ	53.51	54.38	-	52.02	52.04	-
Stanford eng	55.02	53.66	-	53.41	52.74	-
GDep	-	55.70	-	-	54.37	-
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 six parsers in three different formats on the development data set. Results without dependency information are shown as baselines. The results with the GENIA treebank (converted into PTB and PAS) are shown for comparison. 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		
	SD	CoNLL	PAS	SD	CoNLL	PAS
Bikel	53.41 (+0.12)	53.92 (+0.70)	-	51.59 (+0.19)	52.21 (+0.94)	-
Stanford WSJ	53.03 (-0.48)	54.52 (+0.14)	-	51.43 (-0.59)	52.60 (-0.14)	-
Stanford eng	54.48 (-0.54)	54.02 (+0.36)	-	52.88 (-0.53)	52.28 (+0.24)	-
GDep	-	54.97 (-0.73)	-	-	53.71 (-0.66)	-
MC	54.22 (-1.38)	55.24 (-0.77)	-	52.73 (-1.21)	53.42 (-1.09)	-
C&C	54.64 (-1.45)	-	-	52.98 (-1.29)	-	-
Enju	53.74 (-1.74)	**55.66** (-0.08)	55.23 (-1.34)	52.29 (-1.77)	**53.97** (-0.40)	53.69 (-1.62)
GENIA	55.79 (-0.55)	55.64 (-0.45)	56.42 (-1.52)	54.17 (-0.87)	53.83 (-0.74)	55.34 (-1.06)

Table 4: Comparison of F-score results with six parsers in three different dependency structures (without the dependency types) on the development data set. The changes from Table 3 are shown.

	Simple	Binding	Regulation	All
		Task 1		
Ours	**66.84** / **78.22** / **72.08**	48.70 / 52.65 / 50.60	**38.48** / **55.06** / **45.30**	**50.13** / **64.16** / **56.28**
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
Baseline	62.94 / 68.38 / 65.55	48.41 / 34.50 / 40.29	29.40 / 40.00 / 33.89	43.93 / 50.11 / 46.82
		Task 2		
Ours	**65.43** / **75.56** / **70.13**	**46.42** / **50.31** / **48.29**	**38.18** / **54.45** / **44.89**	**49.20** / **62.57** / **55.09**
Riedel	N/A	22.35 / 46.99 / 30.29	25.75 / 40.75 / 31.56	35.86 / 54.08 / 43.12
Baseline	60.88 / 63.78 / 62.30	44.99 / 31.78 / 37.25	29.07 / 39.52 / 33.50	42.62 / 47.84 / 45.08

Table 5: Comparison of Recall / Precision / F-score results on the test data set. Results on simple, binding, regulation, and all events are shown. GDep and Enju with PAS are used. Results by Miwa et al. (2010b), Björne et al. (2009), Riedel et al. (2009), and Baseline for Task 1 and Task 2 are shown for comparison. Baseline results are produced by removing dependency information from the parse results of GDep and Enju. The best score in each result is shown in bold.

tem.

6 Related Work

Many approaches for parser comparison have been proposed, and most comparisons have used gold treebanks with intermediate formats (Clegg and Shepherd, 2007; Pyysalo et al., 2007). Parser comparison has also been proposed on specific tasks such as unbounded dependencies (Rimell et al., 2009) and textual entailment (Önder Eker, 2009)[7]. Among them, 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

[7] http://pete.yuret.com/

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.

7 Conclusion

We compared six parsers and three formats on a bio-molecular event extraction task with a state-of-the-art event extraction system from two different aspects: dependency-based intrinsic evaluation and task-based extrinsic evaluation. The specific task considered was the BioNLP shared task, allowing the use of the GENIA treebank as a gold standard parse reference. Five of the six considered parsers were applied using biomedical models trained on the GENIA treebank, and they were found to produce similar performance. The comparison of the parsers from two aspects showed slightly different results, and and the dependency representations have advantages and disadvantages for the event extraction task.

The contributions of this paper are 1) the comparison of intrinsic and extrinsic evaluation on several commonly used parsers with a state-of-the-art system, and 2) demonstration of the limitation and possibility of the parser and system improvement on the task. One limitation of this study is that the comparison between the parsers is not perfect, as the parsers are used with the provided models, 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 comparison in the present evaluation is limited to the dependency representation. As future work, it would be informative to extend the comparison to other syntactic representation, 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

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

Björne, Jari, 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.

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

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

de Marneffe, Marie-Catherine, 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*.

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

Kim, Jin-Dong, 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.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.

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

Miwa, Makoto, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. 2010a. A comparative study of syntactic parsers for event extraction. In *BioNLP2010: Proceedings of the Workshop on BioNLP*, Uppsala, Sweden, July.

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

Miyao, Yusuke, 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.

Önder Eker. 2009. Parser evaluation using textual entailments. Master's thesis, Boğaziçi Üniversitesi, August.

Pyysalo, Sampo, 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.

Riedel, Sebastian, 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.

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

Rimell, Laura, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 813–821, Singapore, August. Association for Computational Linguistics.

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

Tateisi, Yuka, 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.

Tsuruoka, Yoshimasa, 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 Bozanis, Panayiotis and Elias N. Houstis, editors, *Panhellenic Conference on Informatics*, volume 3746 of *Lecture Notes in Computer Science*, pages 382–392. Springer.