Event Extraction as Dependency Parsing for BioNLP 2011

David McClosky, Mihai Surdeanu, and Christopher D. Manning
Department of Computer Science
Stanford University
Stanford, CA 94305
{mcclosky,mihais,manning}@stanford.edu

Abstract

We describe the Stanford entry to the BioNLP 2011 shared task on biomolecular event extraction (Kim et al., 2011a). Our framework is based on the observation that event structures bear a close relation to dependency graphs. We show that if biomolecular events are cast as these pseudosyntactic structures, standard parsing tools (maximum-spanning tree parsers and parse rerankers) can be applied to perform event extraction with minimum domain-specific tuning. The vast majority of our domain-specific knowledge comes from the conversion to and from dependency graphs. Our system performed competitively, obtaining 3rd place in the Infectious Diseases track (50.6% *f*-score), 5th place in Epigenetics and Post-translational Modifications (31.2%), and 7th place in Genia (50.0%). Additionally, this system was part of the combined system in Riedel et al. (2011) to produce the highest scoring system in three out of the four event extraction tasks.

1 Introduction

The distinguishing aspect of our approach is that by casting event extraction as a dependency parsing, we take advantage of standard parsing tools and techniques rather than creating special purpose frameworks. In this paper, we show that with minimal domain-specific tuning, we are able to achieve competitive performance across the three event extraction domains in the BioNLP 2011 shared task.

At the heart of our system[1] is an off-the-shelf dependency parser, MSTParser[2] (McDonald et al., 2005; McDonald and Pereira, 2006), extended with event extraction-specific features and bookended by conversions to and from dependency trees. While features in MSTParser must be edge-factored and thus fairly local (e.g., only able to examine a portion of each event at once), decoding is performed globally allowing the parser to consider trade-offs. Furthermore, as MSTParser can use n-best decoders, we are able to leverage a reranker to capture global features to improve accuracy.

In §2, we provide a brief overview of our framework. We describe specific improvements for the BioNLP 2011 shared task in §3. In §4, we present detailed results of our system. Finally, in §5 we give some directions for future work.

2 Event Parsing

Our system includes three components: (1) anchor detection to identify and label event anchors, (2) event parsing to form candidate event structures by linking entities and event anchors, and (3) event reranking to select the best candidate event structure. As the full details on our approach are described in McClosky et al. (2011), we will only provide an outline of our methods here along with additional implementation notes.

Before running our system, we perform basic preprocessing on the corpora. Sentences need to be segmented, tokenized, and parsed syntactically. We use custom versions of these (except for Infectious Diseases where we use those from Stenetorp et al. (2011)). To ease event parsing, our

[1] nlp.stanford.edu/software/eventparser.shtml

[2] http://sourceforge.net/projects/mstparser/

Proceedings of BioNLP Shared Task 2011 Workshop, pages 41–45,
Portland, Oregon, USA, 24 June, 2011.

tokenizations are designed to split off suffixes which are often event anchors. For example, we split the token *RelA-induced* into the two tokens *RelA* and *induced*[3] since *RelA* is a protein and *induced* an event anchor. If this was a single token, our event parser would be unable to link them since it cannot predict self-loops in the dependency graph. For syntactic parsing, we use the self-trained biomedical parsing model from McClosky (2010) with the Charniak and Johnson (2005) reranking parser. We use its actual constituency tree, the dependency graph created by applying head percolation rules, and the Stanford Dependencies (de Marneffe and Manning, 2008) extracted from the tree (collapsed and uncollapsed).

Anchor detection uses techniques inspired from named entity recognition to label each token with an event type or *none*. The features for this stage are primarily drawn from Björne et al. (2009). We reduce multiword event anchors to their syntactic head.[4] We classify each token independently using a logistic regression classifier with L_2 regularization. By adjusting a threshold parameter, we can adjust the balance between precision and recall. We choose to heavily favor recall (i.e., overgenerate event anchors) as the event parser can drop extraneous anchors by not attaching any arguments to them.

The event anchors from anchor detection and the included entities (`.t1` files) form a "reduced" sentence, which becomes the input to event parsing. Thus, the only words in the reduced sentence are tokens believed to directly take part in events. Note, though, that we use the original "full" sentence (including the various representations of its syntactic parse) for feature generation. For full details on this process, see McClosky et al. (2011). As stated before, this stage consists of MSTParser with additional event parsing features. There are four decoding options for MSTParser, depending on (a) whether features are first- or second-order and (b) whether graphs produced are projective or non-projective. The projective decoders have complete n-best implementations whereas their non-projective counterparts are approximate. Nevertheless, these four decoders constitute slightly different views of the same data and can be combined inside the reranking framework. After decoding, we convert parses back to event structures. Details on this critical step are given in McClosky et al. (2011).

Event reranking, the final stage of our system, receives an n-best list of event structures from each decoder in the event parsing step. The reranker can use any global features of an event structure to rescore it and outputs the highest scoring structure. This is based on parse reranking (Ratnaparkhi, 1999; Collins, 2000) but uses features on event structures instead of syntactic constituency structures. We used Mark Johnson's `cvlm` estimator[5] (Charniak and Johnson, 2005) when learning weights for the reranking model. Since the reranker can incorporate the outputs from multiple decoders, we use it as an ensemble technique as in Johnson and Ural (2010).

3 Extensions for BioNLP 2011

This section outlines the changes between our BioNLP 2011 shared task submission and the system described in McClosky et al. (2011). The main differences are that all dataset-specific portions of the model have been factored out to handle the expanded Genia (GE) dataset (Kim et al., 2011b) and the new Epigenetics and Post-translational Modifications (EPI) and Infectious Diseases (ID) datasets (Ohta et al., 2011; Pyysalo et al., 2011, respectively). Other changes are relatively minor but documented here as implementation notes.

Several improvements were made to anchor detection, improving its accuracy on all three domains. The first is the use of distributional similarity features. Using a large corpus of abstracts from PubMed (30,963,886 word tokens of 335,811 word types), we cluster words by their syntactic contexts and morphological contents (Clark, 2003). We used the Ney-Essen clustering model with morphology to produce 45 clusters. Using these clusters, we extended the feature set for anchor detection from McClosky et al. (2011) as follows: for each lexicalized feature we create an equivalent feature where the corresponding word is replaced by its cluster ID. This yielded consistent improvements of at least 1 percentage point in both anchor detection and event

[3] The dash is removed since a lone dash would further confuse the syntactic parser.

[4] This does not affect performance if the approximate scorer is used, but it does impact scores if exact matching of anchor boundaries is imposed.

[5] `http://github.com/BLLIP/bllip-parser`

extraction in the development partition of the GE dataset.

Additionally, we improved the head percolation rules for selecting the head of each multiword event anchor. The new rules prohibit determiners and prepositions from being heads, instead preferring verbs, then nouns, then adjectives. There is also a small stop list to prohibit the selection of certain verbs (“has”, “have”, “is”, “be”, and “was”).

In event parsing, we used the *morpha* lemmatizer (Minnen et al., 2001) to stem words instead of simply lowercasing them. This generally led to a small but significant improvement in event extraction across the three domains. Additionally, we do not use the feature selection mechanism described in McClosky et al. (2011) due to time restrictions. It requires running all parsers twice which is especially cumbersome when operating in a round-robin frame (as is required to train the reranker).

Also, note that our systems were only trained to do Task 1 (or “core”) roles for each dataset. This was due to time restrictions and not system limitations.

3.1 Adapting to the Epigenetics track

For the EPI dataset, we adjusted our postprocessing rules to handle the CATALYSIS event type. Similar to REGULATION events in GE, CATALYSIS events do not accept multiple CAUSE arguments. We handle this by replicating such CATALYSIS events and assigning each new event a different CAUSE argument. To adapt the ontology features in the parser (McClosky et al., 2011, §3.3), we created a supertype for all non-CATALYSIS events since they behave similarly in many respects.

There are several possible areas for improvement in handling this dataset. First, our internal implementation of the evaluation criteria differed from the online scorer, sometimes by up to 6% *f*-score. As a result, the reranker optimized a noisy version of the evaluation criteria and potentially could have performed better. It is unclear why our evaluator scored EPI structures differently (it replicated the scores for GE) but it is worthy of investigation. Second, due to time constraints, we did not transfer the parser or reranker consistency features (e.g., non-REGULATION events should not take events as arguments) or the type ontology in the reranker to the EPI dataset. As a result, our results describe our system with incomplete domain-specific knowledge.

3.2 Adapting to the Infectious Diseases track

Looking only at event types and their arguments, ID is similar to GE. As a result, much of our domain-specific processing code for this dataset is based on code for GE. The key difference is that the GE post-processing code removes event anchors with zero arguments. Since ID allows PROCESS events to have zero or one anchors, we added this as an exception. Additionally, the ID dataset includes many nested entities, e.g., two-component system entities contain two other entities within their span. In almost all of these cases, only the outermost entity takes part in an event. To simplify processing, we removed all nested entities. Any events attaching to a nested entity were reattached to its outermost entity.

Given the similarities with GE, we explored simple domain adaptation by including the gold data from GE along with our ID training data. To ensure that the GE data did not overwhelm the ID data, we tried adding multiple copies of the ID data (see Table 1 and the next section).

As in EPI, we adjusted the type ontology in the parser for this dataset. This included “core entities” (as defined by the task) and a “PROTEIN-or-REGULON-OPERON” type (the type of arguments for GENE EXPRESSION and TRANSCRIPTION events). Also as in EPI, the reranker did not use the updated type ontology.

4 Results

For ID, we present experiments on merging GE with ID data (Table 1). Since GE is much larger than ID, we experimented with replicating the ID training partition. Our best performance came from training on three copies of the ID data and the training and development sections of GE. However, as the table shows, performance is stable for more than two copies of the ID data. Note that for this shared task we simply merged the two domains. We did not implement any domain adaptation techniques (e.g., labeling features based on the domain they come from (Daumé III, 2007)).

Table 2 shows the performance of the various parser decoders and their corresponding rerankers. The last line in each domain block lists the score of the reranker that uses candidates produced by all decoders. This reranking model always outperforms the best individual parser. Furthermore, the reranking models on top of individual decoders help in all but one situation (ID – 2N decoder). To our knowledge, our approach is the first to show that reranking with features generated from global event structure helps event extraction. Note that due to approximate 2N decoding in MSTParser, this decoder does not produce true n-best candidates and generally outputs only a handful of unique parses. Because of this, the corresponding rerankers suffer from insufficient training data and hurt performance in ID.

Finally, in Table 3, we give our results and ranking on the official test sets. Our results are 6 f points lower than the best submission in GE and EPI and 5 points lower in ID. Considering that the we used generic parsing tools with minimal customization (e.g., our parsing models cannot extract directed acyclic graph structures, which are common in this data), we believe these results are respectable.

5 Conclusion

Our participation in the BioNLP shared task proves that standard parsing tools (i.e., maximum-spanning tree parsers, parse rerankers) can be successfully used for event extraction. We achieved this by converting the original event structures to a pseudo-syntactic representation, where event arguments appear as modifiers to event anchors. Our analysis indicates that reranking always helps, which proves that there is merit in modeling non-local information in biomolecular events. To our knowledge, our approach is the first to use parsing models for biomedical event extraction.

During the shared task, we adapted our system previously developed for the 2009 version of the Genia dataset. This process required minimal effort: we did not add any new features to the parsing model; we added only two domain-specific post-processing steps (i.e., we allowed events without arguments in ID and we replicated CATALYSIS events with multiple CAUSE arguments in EPI). Our system's robust performance in all domains proves that our approach is portable.

A desired side effect of our effort is that we can easily incorporate any improvements to parsing models (e.g., parsing of directed acyclic graphs, dual decomposition, etc.) in our event extractor.

Model	Prec	Rec	f-score
ID	**59.3**	38.0	46.3
(ID×1) + GE	52.0	40.2	45.3
(ID×2) + GE	52.4	41.7	46.4
(ID×3) + GE	54.8	**45.0**	**49.4**
(ID×4) + GE	55.2	43.8	48.9
(ID×5) + GE	55.1	44.7	**49.4**

Table 1: Impact of merging several copies of ID training with GE training and development. Scores on ID development data (2N parser only).

Decoder(s)	Parser	Reranker
1P	49.0	49.4
2P	49.5	50.5
1N	**49.9**	50.2
2N	46.5	47.9
All	—	**50.7** ∗

(a) Genia results (task 1)

Decoder(s)	Parser	Reranker
1P	62.3	63.3
2P	62.2	63.3
1N	**62.9**	**64.6** ∗
2N	60.8	63.8
All	—	64.1

(b) Epigenetics results (core task)

Decoder(s)	Parser	Reranker
1P	46.0	48.5
2P	47.8	49.8
1N	48.5	49.4
2N	**49.4**	48.8
All	—	**50.2** ∗

(c) Infectious Diseases results (core task)

Table 2: Results on development sections in BioNLP f-scores. "∗" indicates the submission model for each domain.

Domain (task)	Prec	Rec	f-score	Ranking
GE (task 1)	61.1	42.4	50.0	7th
EPI (core)	70.2	56.9	62.8	5th
ID (core)	55.9	46.3	50.6	3rd

Table 3: BioNLP f-scores on the final test set.

Acknowledgments

We would like to thank the BioNLP shared task organizers for an enjoyable and interesting task and their quick responses to questions. We would also like to thank Sebastian Riedel for many interesting discussions. We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181.

References

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 2009 Workshop Companion Volume for Shared Task*, pages 10–18, Boulder, Colorado, June. Association for Computational Linguistics.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *ACL*. The Association for Computer Linguistics.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth Annual Meeting of the European Association for Computational Linguistics (EACL)*, pages 59–66.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML 2000)*, pages 175–182, Stanford, California.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Conference of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed hierarchies representation. In *Proceedings of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation*.

Mark Johnson and Ahmet Engin Ural. 2010. Reranking the berkeley and brown parsers. In *Proceedings of the HLT: North American Chapter of the ACL (HLT-NAACL)*, pages 665–668. Association for Computational Linguistics, June.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Jun'ichi Tsujii. 2011a. Overview of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011b. Overview of the Genia Event task in BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

David McClosky, Mihai Surdeanu, and Chris Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the Association for Computational Linguistics: Human Language Technologies 2011 Conference (ACL-HLT'11), Main Conference*, Portland, Oregon, June.

David McClosky. 2010. *Any Domain Parsing: Automatic Domain Adaptation for Parsing*. Ph.D. thesis, Computer Science Department, Brown University.

Ryan T. McDonald and Fernando C. N. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*. The Association for Computer Linguistics.

Ryan T. McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT/EMNLP*. The Association for Computational Linguistics.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(03):207–223.

Tomoko Ohta, Sampo Pyysalo, and Jun'ichi Tsujii. 2011. Overview of the Epigenetics and Post-translational Modifications (EPI) task of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. 2011. Overview of the Infectious Diseases (ID) task of BioNLP Shared Task 2011. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34(1-3):151–175.

Sebastian Riedel, David McClosky, Mihai Surdeanu, Andrew McCallum, and Christopher D. Manning. 2011. Model Combination for Event Extraction in BioNLP 2011. In *BioNLP 2011 Shared Task*.

Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. 2011. BioNLP Shared Task 2011: Supporting Resources. In *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task*, Portland, Oregon, June. Association for Computational Linguistics.