Robust Biomedical Event Extraction with Dual Decomposition and Minimal Domain Adaptation

Sebastian Riedel **Andrew McCallum**
Department of Computer Science
University of Massachusetts, Amherst
{riedel,mccallum}@cs.umass.edu

Abstract

We present a joint model for biomedical event extraction and apply it to four tracks of the BioNLP 2011 Shared Task. Our model decomposes into three sub-models that concern (a) event triggers and outgoing arguments, (b) event triggers and incoming arguments and (c) protein-protein bindings. For efficient decoding we employ dual decomposition. Our results are very competitive: With minimal adaptation of our model we come in second for two of the tasks—right behind a version of the system presented here that includes predictions of the Stanford event extractor as features. We also show that for the Infectious Diseases task using data from the Genia track is a very effective way to improve accuracy.

1 Introduction

This paper presents the UMass entry to the BioNLP 2011 shared task (Kim et al., 2011a). We introduce a simple joint model for the extraction of biomedical events, and show competitive results for four tracks of the competition. Our model subsumes three tractable sub-models, one for extracting event triggers and outgoing edges, one for event triggers and incoming edges and one for protein-protein bindings. Fast and accurate joint inference is provided by combining optimizing methods for these three submodels via dual decomposition (Komodakis et al., 2007; Rush et al., 2010). Notably, our model constitutes the first joint approach that explicitly predicts which protein should share the same binding event. So far this has either been done through postprocessing heuristics (Björne et al., 2009; Riedel et al., 2009; Poon and Vanderwende, 2010), or through a local classifier at the end of a pipeline (Miwa et al., 2010).

Our model is very competitive. For Genia (GE) Task 1 (Kim et al., 2011b) we achieve the second-best results. In addition, the best-performing FAUST system (Riedel et al., 2011) is a variant of the model presented here. Its advantage stems from the fact that it uses predictions of the Stanford system (McClosky et al., 2011a; McClosky et al., 2011b), and hence performs model combination. The same holds for the Infectious Diseases (ID) track (Pyysalo et al., 2011), where we come in as second right behind the FAUST system. For the Epigenetics and Post-translational Modifications (EPI) track (Ohta et al., 2011) we achieve the 4th rank, partly because we did not aim to extract speculations, negations or cellular locations. Finally, for Genia Task 2 we rank 3rd—with the 1st rank achieved by the FAUST system.

In the following we will briefly describe our model and inference algorithm, as far as this is possible in limited space. Then we show our results on the three tasks and conclude. Note we will assume familiarity with the task, and refer the reader to the shared task overview paper for more details.

2 Biomedical Event Extraction

Our goal is to extract biomedical events as shown in figure 1a). To formulate the search for such structures as an optimization problem, we represent structures through a set of binary variables. Our representation is inspired by previous work (Riedel et al., 2009; Björne et al., 2009) and based on a projection of events to a labelled graph over tokens in the

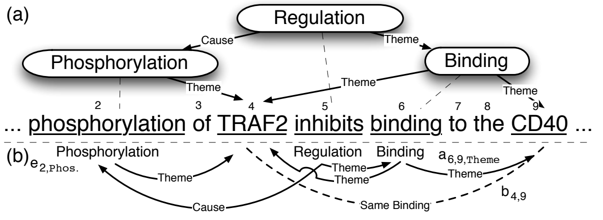

Figure 1: (a) sentence with target event structure; (b) projection to labelled graph.

sentence, as seen figure 1b).

We will first present some basic notation to simplify our exposition. For each sentence $\mathbf{x}$ we have a set candidate trigger words $\text{Trig}(\mathbf{x})$, and a set of candidate proteins $\text{Prot}(\mathbf{x})$. We will generally use the indices i and l to denote members of $\text{Trig}(\mathbf{x})$, the indices p, q for members of $\text{Prot}(\mathbf{x})$ and the index j for members of $\text{Cand}(\mathbf{x}) \stackrel{\text{def}}{=} \text{Trig}(\mathbf{x}) \cup \text{Prot}(\mathbf{x})$.

We label each candidate trigger i with an event Type $t \in \mathcal{T}$ (with $\mathtt{None} \in \mathcal{T}$), and use the binary variable $e_{i,t}$ to indicate this labeling. We use binary variables $a_{i,l,r}$ to indicate that between i and l there is an edge labelled $r \in \mathcal{R}$ (with $\mathtt{None} \in \mathcal{R}$).

The representation so far has been used in previous work (Riedel et al., 2009; Björne et al., 2009). Its shortcoming is that it does not capture whether two proteins are arguments of the same binding event, or arguments of two binding events with the same trigger. To overcome this problem, we introduce binary "same Binding" variables $b_{p,q}$ that are active whenever there is a binding event that has both p and q as arguments. Our inference algorithm will also need, for each trigger i and protein pair p, q, a binary variable $t_{i,p,q}$ that indicates that at i there is a binding event with arguments p and q. All $t_{i,p,q}$ are summarized in $\mathbf{t}$.

Constructing events from solutions $(\mathbf{e}, \mathbf{a}, \mathbf{b})$ can be done almost exactly as described by Björne et al. (2009). However, while Björne et al. (2009) group arguments according to ad-hoc rules based on dependency paths from trigger to argument, we simply query the variables $b_{p,q}$.

3 Model

We use the following objective to score the structures we like to extract:

$$s(\mathbf{e}, \mathbf{a}, \mathbf{b}) \stackrel{\text{def}}{=} \sum_{e_{i,t}=1} s_{\text{T}}(i,t) + \sum_{a_{i,j,r}=1} s_{\text{R}}(i,j,r) + \sum_{b_{p,q}=1} s_{\text{B}}(p,q)$$

with local scoring functions $s_{\text{T}}(i,t) \stackrel{\text{def}}{=} \langle \mathbf{w}_{\text{T}}, \mathbf{f}_{\text{T}}(i,t) \rangle$, $s_{\text{R}}(i,j,r) \stackrel{\text{def}}{=} \langle \mathbf{w}_{\text{R}}, \mathbf{f}_{\text{R}}(i,j,r) \rangle$ and $s_{\text{B}}(p,q) \stackrel{\text{def}}{=} \langle \mathbf{w}_{\text{B}}, \mathbf{f}_{\text{B}}(p,q) \rangle$.

Our model scores all parts of the structure in isolation. It is a joint model due to the three types of constraints we enforce. The first type acts on trigger labels and their *outgoing* edges. It includes constraints such as "an active label at trigger i requires at least one active outgoing `Theme` argument". The second type enforces consistency between trigger labels and their *incoming* edges. That is, if an incoming edge has a label that is not `None`, the trigger must not be labelled `None` either. The third type of constraints ensures that when two proteins p and q are part of the same binding (as indicated by $b_{p,q} = 1$), there needs to be a binding event at some trigger i that has p and q as arguments. We will denote the set of structures $(\mathbf{e}, \mathbf{a}, \mathbf{b})$ that satisfy all above constraints as $\mathcal{Y}$.

To learn $\mathbf{w}$ we choose the passive-aggressive online learning algorithm (Crammer and Singer, 2003). As loss function we apply a weighted sum of false positives and false negative labels and edges. The weighting scheme penalizes false negatives 3.8 times more than false positives.

3.1 Features

For feature vector $\mathbf{f}_{\text{T}}(i,t)$ we use a collection of representations for the token i: word-form, lemma, POS tag, syntactic heads, syntactic children; membership in two dictionaries used by Riedel et al. (2009).For $\mathbf{f}_{\text{R}}(a; i, j, r)$ we use representations of the token pair (i, j) inspired by Miwa et al. (2010) . They contain: labelled and unlabeled n-gram dependency paths; edge and vertex walk features (Miwa et al., 2010), argument and trigger modifiers and heads, words in between (for close distance i and j). For $\mathbf{f}_{\text{B}}(b; p, q)$ we use a small subset of the token pair representations in $\mathbf{f}_{\text{R}}$.

Algorithm 1 Dual Decomposition.

```
require:
    $R$: max. iteration, $\alpha_t$: stepsizes
$t \leftarrow 0$ $\boldsymbol{\lambda} \leftarrow 0$ $\boldsymbol{\mu} \leftarrow 0$
repeat
    $(\bar{\mathbf{e}}, \bar{\mathbf{a}}) \leftarrow \text{bestIncoming}\left(-\boldsymbol{\lambda}\right)$
    $(\mathbf{e}, \mathbf{a}) \leftarrow \text{bestOutgoing}\left(\mathbf{c}^{\text{out}}\left(\boldsymbol{\lambda}, \boldsymbol{\mu}\right)\right)$
    $(\mathbf{b}, \mathbf{t}) \leftarrow \text{bestBinding}\left(\mathbf{c}^{\text{bind}}\left(\boldsymbol{\mu}\right)\right)$
    $\lambda_{i,t} \leftarrow \lambda_{i,t} - \alpha_t\left(e_{i,t} - \bar{e}_{i,t}\right)$
    $\lambda_{i,j,r} \leftarrow \lambda_{i,j,r} - \alpha_t\left(a_{i,j,r} - \bar{a}_{i,j,r}\right)$
    $\mu^{\text{trig}}_{i,j,k} \leftarrow \left[\mu^{\text{trig}}_{i,j,k} - \alpha_t\left(e_{i,\texttt{Bind}} - t_{i,j,k}\right)\right]_+$
    $\mu^{\text{arg1}}_{i,j,k} \leftarrow \left[\mu^{\text{arg1}}_{i,j,k} - \alpha_t\left(a_{i,j,\texttt{Theme}} - t_{i,j,k}\right)\right]_+$
    $\mu^{\text{arg2}}_{i,j,k} \leftarrow \left[\mu^{\text{arg2}}_{i,j,k} - \alpha_t\left(a_{i,k,\texttt{Theme}} - t_{i,j,k}\right)\right]_+$
    $t \leftarrow t + 1$
until no $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ changed or $t > R$
return$(\mathbf{e}, \mathbf{a}, \mathbf{b})$
```

3.2 Inference

Inference in our model amounts to solving

$$\arg \max_{(\mathbf{e},\mathbf{a},\mathbf{b}) \in \mathcal{Y}} s\left(\mathbf{e}, \mathbf{a}, \mathbf{b}\right). \tag{1}$$

Our approach to finding the maximizer is dual decomposition (Komodakis et al., 2007; Rush et al., 2010), a technique that allows us to exploit efficient search algorithms for tractable substructures of our problem. We divide the problem into three sub-problems: (1) finding the highest-scoring trigger labels and edges $(\mathbf{e}, \mathbf{a})$ such that constraints on triggers and their outgoing edges are fulfilled; (2) finding the highest-scoring trigger labels and edges $(\bar{\mathbf{e}}, \bar{\mathbf{a}})$ such that constraints on triggers and their incoming edges are fulfilled; (3) finding the highest-scoring pairs of proteins $\mathbf{b}$ to appear in the same binding, and make binding event trigger decisions $\mathbf{t}$ for these. Due to space constraints we only state that the first two problems can be solved exactly in $O\left(n^2 + nm\right)$ time while the last needs $O\left(m^2 n\right)$. Here n is the number of trigger candidates and m the number of proteins.

The subroutines to solve these three sub-problems are combined in algorithm 1—an instantiation of subgradient descent on the dual of an LP relaxation of problem 1. In the first three steps in the main loop of this algorithm, the individual sub-problems are solved. Note that to each subroutine a parameter is passed. For example, when finding the structure $(\bar{\mathbf{e}}, \bar{\mathbf{a}})$ that maximizes the objective under the incoming edge constraints, we pass the parameter $-\boldsymbol{\lambda}$. This parameter represents a set of *penalties* to be added to the objective used for the subproblem. In this case we have penalties $-\lambda_{i,e}$ to be added to the scores of trigger-label pairs (i, e), and penalties $-\lambda_{i,j,r}$ to be added for labelled edges $i \xrightarrow{r} j$.

One way to understand dual decomposition is as iterative tuning of the penalties such that eventually all individual solutions are consistent with each other. In our case this would mean, among other things, that the solutions $(\mathbf{e}, \mathbf{a})$ and $(\bar{\mathbf{e}}, \bar{\mathbf{a}})$ are identical. This tuning happens in the second part of the main loop which updates the *dual variables* $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. We see, for example, how the penalties $\lambda_{i,e}$ are decreased by $e_{i,e} - \bar{e}_{i,e}$ scaled by a step-size α_t. Effectively this change to $\lambda_{i,e}$ will decrease the score of $\bar{e}_{i,e}$ within bestIn $\left(-\boldsymbol{\lambda}\right)$ by α_t if $\bar{e}_{i,e}$ was true while $e_{i,e}$ was false in the current solutions.[1] If $\bar{e}_{i,e}$ was false but $e_{i,e}$ was true, the score is increased by α_t. If both agree, no change is needed.

Consistency between solutions also means that the binding decisions in $\mathbf{b}$ and $\mathbf{t}$ are consistent with the rest of the solution. This is achieved in algorithm 1 through tuning of the dual variables $\boldsymbol{\mu}$ but we omit details for brevity. For completeness we state how the penalties used for solving the other subproblems are set based on the dual variables $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. We set $\mathbf{c}^{\text{out}}_{i,t}\left(\boldsymbol{\lambda}, \boldsymbol{\mu}\right) \stackrel{\text{def}}{=} \lambda_{i,t} + \delta_{t,\texttt{Bind}} \sum_{p,q} \mu^{\text{trig}}_{i,p,q}$; for the case that $j \in \text{Prot}\left(\mathbf{x}\right)$ we get $\mathbf{c}^{\text{out}}_{i,j,r}\left(\boldsymbol{\lambda}, \boldsymbol{\mu}\right) \stackrel{\text{def}}{=} \lambda_{i,j,r} + \sum_p \mu^{\text{arg1}}_{i,j,p} + \sum_q \mu^{\text{arg2}}_{i,q,j}$, otherwise $\mathbf{c}^{\text{out}}_{i,j,r}\left(\boldsymbol{\lambda}, \boldsymbol{\mu}\right) \stackrel{\text{def}}{=} \lambda_{i,j,r}$. For bestBind $\left(\mathbf{c}\right)$ we set $\mathbf{c}^{\text{bind}}_{i,p,q}\left(\boldsymbol{\mu}\right) = -\mu^{\text{trig}}_{i,p,q} - \mu^{\text{arg1}}_{i,,p,q} - \mu^{\text{arg2}}_{i,,p,q}$.

3.3 Preprocessing

After basic tokenization and sentence segmentation, we generate a set of protein head tokens $\text{Prot}\left(\mathbf{x}\right)$ for each sentence $\mathbf{x}$ based on protein span definitions from the shared task. To ensure tokens contain not more than one protein we split them at protein boundaries. Parsing is performed using the Charniak-Johnson parser (Charniak and Johnson, 2005) with the self-trained biomedical parsing

[1] We refer to Koo et al. (2010) for details on how to set α_t.

	SVT	BIND	REG	TOT
Task 1	73.5	48.8	43.8	55.2
Task 1 (abst.)	71.5	50.8	45.5	56.1
Task 1 (full)	79.2	44.4	40.1	53.1
Task 2	71.4	38.6	39.1	51.0

Table 1: Results for the GE track, task 1 and 2; abst.=abstract; full=full text.

model of McClosky and Charniak (2008). Finally, based on the set of trigger words in the training data, we generate a set of candidate triggers Trig $(\mathbf{x})$.

4 Results

We apply the same model to the GE, ID and EPI tracks, with minor modifications in order to deal with the different event type sets $\mathcal{T}$ and role sets $\mathcal{R}$ of each track. Training and testing together took between 30 (EPI) to 120 (GE) minutes using a single-core implementation.

4.1 Genia

Our results for GE task 1 and 2 can be seen in table 1. We also show results for abstracts only (abst.), and for full text only (full). Note that binding events (BIND) and general regulation events (REG) seem to be harder to extract in full text. Somewhat surprisingly, for simple events (SVT) the opposite holds. We also like to point out that for full text extraction we rank first—the second best FAUST system achieves an F1 score of 52.67.

4.2 Infectious Diseases

The Infectious Diseases track differs from the Genia track in two important ways. First, it introduces the event type `Process` that is allowed to have no arguments at all. Second, it comes with significantly less training data (152 vs 908 documents). We can accommodate the first difference by making simple changes in our inference algorithms. For example, for `Process` events we do not force the algorithm to pick a `Theme` argument.

To compensate for the lack of training data we simply add data from the GE track. This is reasonable because annotations overlap quite significantly. In table 2 we show the impact of mixing different amounts of ID data (I) and GE data (G) into the training set. We point out that adding the ID training set twice, and the GENIA set once, leads to the best performance (I/G=2/1). Remarkably, the F1 score for `Process` increases by including data, although this data does not include any such events. This may stem from a shared model of `None` arguments that is improved with more data.

	I/G	BIND	REG	PRO	TOT
DEV	1/0	18.6	27.1	34.3	41.5
DEV	0/1	18.2	26.8	0.00	35.5
DEV	1/1	20.0	33.1	49.3	47.2
DEV	2/1	20.0	34.5	52.0	48.5
TEST	2/1	**34.6**	**46.4**	**62.3**	**53.4**

Table 2: ID results for different amounts of ID (I) and (G) training data.

4.3 Epigenetics and Post-translational Modifications

For this track a different set of events is to be predicted. However, it is straightforward to adapt our model and algorithms to this setting. For brevity we only report our total results here and omit a table with details. The first metric (ALL) includes negation, speculation and cellular location targets. We omitted these in our model and hence our result of 33.52 F1 is relatively weak. For the metric that neglects these aspects (CORE), we achieve 64.15 F1 and come in 4th. Note that in this metric the FAUST system, based on the model presented here, comes in as very close second.

5 Conclusion

We have presented a robust joint model for event extraction from biomedical text that performs well across all tasks. Remarkably, no feature set or parameter tuning was necessary to achieve this. We also show substantial improvements for the ID task by adding GENIA data into the training set.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval. The University of Massachusetts gratefully acknowledges the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

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 Natural Language Processing in Biomedicine NAACL 2009 Workshop (BioNLP '09)*, pages 10–18, Morristown, NJ, USA. Association for Computational Linguistics.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pages 173–180.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

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.

Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. 2007. Mrf optimization via dual decomposition: Message-passing revisited. In *In ICCV*.

Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of the 46rd Annual Meeting of the Association for Computational Linguistics (ACL '08)*.

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

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011b. Event extraction as dependency parsing in BioNLP 2011. In *BioNLP 2011 Shared Task*.

Makoto Miwa, Rune Saetre, Jin-Dong D. Kim, and Jun'ichi Tsujii. 2010. Event extraction with complex event classification using rich features. *Journal of bioinformatics and computational biology*, 8(1):131–146, February.

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.

Hoifung Poon and Lucy Vanderwende. 2010. Joint Inference for Knowledge Extraction from Biomedical Literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 813–821, Los Angeles, California, 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.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A markov logic approach to bio-molecular event extraction. In *Proceedings of the Natural Language Processing in Biomedicine NAACL 2009 Workshop (BioNLP '09)*, pages 41–49.

Sebastian Riedel, David McClosky, Mihai Surdeanu, Christopher D. Manning, and Andrew McCallum. 2011. Model combination for event extraction in BioNLP 2011. In *BioNLP 2011 Shared Task*.

Alexander M. Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *In Proc. EMNLP*.