Biomedical Event Extraction by Multi-class Classification of Pairs of Text Entities

Xiao Liu, Antoine Bordes, Yves Grandvalet

Biomedical Event Extraction by Multi-class Classification of Pairs of Text Entities

Xiao Liu **Antoine Bordes** **Yves Grandvalet**
Université de Technologie de Compiègne & CNRS
Heudiasyc UMR 7253, Rue Roger Couttolenc, CS 60319
60203 Compiègne Cedex, FRANCE
firstname.lastname@hds.utc.fr

Abstract

This paper describes the HDS4NLP entry to the BioNLP 2013 shared task on biomedical event extraction. This system is based on a pairwise model that transforms trigger classification in a simple multi-class problem in place of the usual multi-label problem. This model facilitates inference compared to global models while relying on richer information compared to usual pipeline approaches. The HDS4NLP system ranked 6th on the Genia task (43.03% f-score), and after fixing a bug discovered after the final submission, it outperforms the winner of this task (with a f-score of 51.15%).

1 Introduction

Huge amounts of electronic biomedical documents, such as molecular biology reports or genomic papers are generated daily. Automatically organizing their content in dedicated databases enables advanced search and ease information retrieval for practitioners and researchers in biology, medicine or other related fields. Nowadays, these data sources are mostly in the form of unstructured free text, which is complex to incorporate into databases. Hence, many research events are organized around the issue of automatically extracting information from biomedical text. Efforts dedicated to biomedical text are necessary because standard Natural Language Processing tools cannot be readily applied to extract biomedical events since they involve highly domain-specific jargon and dependencies (Kim et al., 2011).

This paper describes the HDS4NLP entry to one of these challenges, the Genia task (GE) of the BioNLP 2013 shared task. The HDS4NLP system is based on a novel model designed to directly extract events having a pairwise structure (*trigger*, *argument*), in contrast to standard pipeline models which first extract the trigger and then search for the argument. Combining these two steps enables to use more sophisticated event features while largely avoiding error propagation. The model usage is also simple, in the sense that it does not rely on any complex and costly inference process as required by joint global systems based on Integer Linear Programming.

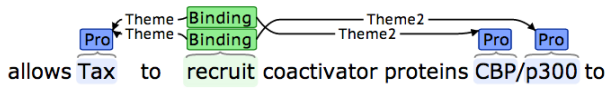

Figure 1: Part of a sentence and corresponding events for the BioNLP 2013 GE task.

The official HDS4NLP entry was only ranked 6th on the GE task (with 43.03% f-score). However, after fixing a bug discovered after the final submission, the HDS4NLP system outperformed the winner of the GE task, with a f-score 51.15% to be compared to the 50.97% of EVEX.

2 BioNLP Genia Task

BioNLP Genia task aims at extracting *event formulas* from text sentences, which are defined as sequences of *tokens* (words, numbers, or symbols). Events are constituted of two elements: an event *trigger* and one or several *arguments*. The event trigger is a sequence of tokens that indicates an event is mentioned in the text. The arguments of an event are participants, which can be proteins, genes or other biomedical events. Figure 1 illustrates the GE task: given 3 proteins "*Tax*", "*CBP*" and "*p300*", one must detect "*recruit*" as an event trigger and then extract two formulas: ("*recruit*", Theme:"*Tax*", Theme2:"*CBP*") and ("*recruit*", Theme:"*Tax*", Theme2:"*p300*"), both with event type *Binding*.

In our work, we process tokens differently depending on whether they are marked as proteins in the annotation or not; the latter are termed *candidate* tokens. A key part of the task is to detect the

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 45–49,
Sofia, Bulgaria, August 9 2013.

trigger tokens among the candidates. The BioNLP 2013 GE task considers 13 types of events, but we only dealt with the 9 types already existing in the 2011 GE task, because there was not enough data on the newly defined event types for proper training or model selection.

Table 1 lists these events and their properties. The 9 event types may be merged into three main groups: the first 5 have a single argument, a *Theme*; the *Binding* event can accept up to two arguments (2 Themes); the last 3 types also accept up to two arguments, a Theme and an optional *Cause*. In the following, we refer to the first 6 types as *non-regulation* events and to the remaining 3 as *regulation* ones.

Event type	Principal arg	Optional arg
Gene_expression	Theme (P)	
Transcription	Theme (P)	
Protein_catabolism	Theme (P)	
Phosphorylation	Theme (P)	
Localization	Theme (P)	
Binding	Theme (P)	Theme2 (P)
Regulation	Theme (E/P)	Cause (E/P)
Positive_regulation	Theme (E/P)	Cause (E/P)
Negative_regulation	Theme (E/P)	Cause (E/P)

Table 1: Main types of events with their arguments (P stands for *Protein*, E for *Event*).

3 Previous Work

The preceding approaches falls into two main categories: pipeline incremental models and joint global methods.

Pipeline approaches (Sætre et al., 2009; Cohen et al., 2009; Björne et al., 2009) are the simplest way to tackle the problem of event extraction. A sequence of classifiers are ran on the text to successively (1) detect non-regulation event triggers, (2) assign them arguments, (3) detect regulation event triggers and (4) assign them arguments. Such systems are relatively easy to set up but suffer from error cascading. Besides, they detect triggers using classifiers solely taking tokens as input, or involve dependency parse information by tree depth other than a concrete potential argument (Björne et al., 2009).

In the corpuses used in 2009 and 2011 for the GE task, some tokens belong to several events of different types; their classification thus requires to solve a multi-label problem. We believe that detecting triggers in isolation breaks the structured problem down to excessively fine-grained subtasks, with contextual information loss that leads to ill-posed problems.

Global approaches (Riedel et al., 2009; McClosky et al., 2011) aim at solving the whole task at once, so as to resolve the drawbacks of pipeline models. In (McClosky et al., 2011), event annotations are converted into pseudo-syntactic representations and the task is solved as a syntactic extraction problem by traditional parsing methods. (Riedel et al., 2009; Riedel and McCallum, 2011a; Riedel et al., 2011; Riedel and McCallum, 2011b) encode the event annotations as latent binary variables indicating the type of each token and the relation between each pair of them (protein or candidate) in a sentence. The state of these variables is predicted by maximizing the global likelihood of an Integer Linear Program.This joint model achieves good performance (winner of the 2011 GE task), but might be overly complicated, as it considers all possible combinations of tokens, even unlikely ones, as potential events together.

4 Pairwise Model

Our new pairwise approach operates at the sentence level. We denote $\mathcal{C}_S = \{e_i\}_i$ the set of candidate tokens, $\mathcal{A}_S = \{a_j\}_j$ the set of candidate arguments in a given sentence S, and the set of event types (augmented by *None*) is denoted $\mathcal{Y}$.

The first step of a pipeline model assigns labels to candidate tokens $e_i \in \mathcal{C}_S$. Instead, our pairwise model addresses the problem of classifying candidate-argument pairs $(e_i, a_j) \in \mathcal{C}_S \times \mathcal{A}_S$. Denoting f_k the binary classifier predicting the event type $k \in \mathcal{Y}$, event extraction is performed by:

$$\forall (e_i, a_j) \in \mathcal{C}_S \times \mathcal{A}_S,\ \ \hat{y}_{ij} = \underset{k \in \mathcal{Y}}{\arg\max}\, f_k(e_i, a_j)\ .$$

Variable $\hat{y}_{ij}$ encodes the event type of the pair made of the candidate token e_i and the argument a_j, an event being actually extracted when $\hat{y}_{ij} \neq None$. For the f_k classifiers, we use Support Vector Machines (SVMs) (using implementation from `scikit-learn.org`) in a one-vs-rest setting. We used procedures from (Duan et al., 2003; Platt, 1999; Tax and Duin, 2002) to combine the outputs of these binary classifiers in order to predict a single class from $\mathcal{Y}$ for each pair (e_i, a_j).

This simple formulation is powerful because classifying a pair (e_i, a_j) as not-*None* jointly detects the event trigger e_i and its argument a_j. For all event types with a single argument, predicting $\hat{y}$ variables directly solves the task. Working on pairs (e_i, a_j) also allows to take into account interactions, in particular through dedicated features

describing the connection between the trigger and its argument (see Section 6). Finally, classifying pairs (e_i, a_j) is conceptually simpler than classifying e_i: the task is a standard classification problem instead of a multi-label problem. Note that entity e_i may still be assigned to several categories through the allocation of different labels to pairs (e_i, a_j) and (e_i, a_k).

Though being rather minimalist, the pairwise structure captures a great deal of trigger-argument interactions, and the simplicity of the structure leads to a straightforward inference procedure. Compared to pipeline models, the main drawback of the pairwise model is to multiply the number of examples to classify by a card($\mathcal{A}_S$) factor. However, SVMs can scale to large numbers of examples and card($\mathcal{A}_S$) is usually low (less than 10).

5 Application to BioNLP Genia Task

For any given sentence, our system sequentially solves a set of 4 pairwise relation extraction problems in the following order:

1. Trigger-Theme pair extraction (**T-T**),
2. Binding-Theme fusion (**B-T**),
3. Regulation-Theme pair extraction (**R-T**),
4. Regulation-Cause assignment (**R-C**).

Steps T-T and R-T are the main event extraction steps because they detect the triggers and one argument. Since some events can accept multiple arguments, we supplement T-T and R-T with steps B-T and R-C, designed to potentially add arguments to events. All steps are detailed below.

Steps T-T & R-T Both steps rely on our pairwise model to jointly extract event triggers, determine their types and corresponding themes. However, they detect different triggers with different potential argument sets: for step T-T, $\mathcal{A}_S$ contains only proteins and $\mathcal{Y}$ = {*Gene_expression*, *Transcription*, *Protein_catabolism*, *Phosphorylation*, *Localization*, *Binding*, *None*}. For step R-T, $\mathcal{A}_S$ contains proteins and all predicted triggers, $\mathcal{Y}$ = {*Regulation*, *Positive_regulation*, *Negative_regulation*, *None*}.

Steps B-T & R-C These steps attempt to assign optional arguments to *Binding* or regulation events detected by T-T or R-T respectively. They proceed similarly. Given an extracted event (e_i, a_j) and a candidate argument set $A_S = \{a_k\}$, all combinations $\{(e_i, a_j, a_k) | k \neq j\}$ are classified by a binary SVM. For B-T, A_S contains all the proteins of the sentence S that were extracted as argument of a *Binding* event by T-T. For R-C, A_S contains all proteins and triggers detected by T-T. In both cases, a post-processing step is used to select the longest combination.

Type	Features
Surface features	Stem
	String after '-'
	String while pruning '-' and/or '/'
	Prefix of token
Semantic features	Lemma from WordNet
	Part-of-speech (POS) tag
	Token annotated as protein

Table 2: Word features.

6 Features

We present here our features and preprocessing.

Candidate set For each sentence S, the set $\mathcal{C}_S$ is built using a trigger gazetteer: candidates are recursively added by searching first the longest tokens sequences from the gazetteer. For candidates with several tokens, a *head* token is selected using an heuristic based on the dependency parse.

Candidate tokens Three types of features are used, either related to the head token, a word window around it, or its parent and child nodes in the dependency tree. Table 2 lists word features.

Proteins The protein name is a useless feature, so the word features of the head token were removed for proteins. Word features of the neighboring tokens and of the parent node in the dependency tree were still included. Proteins are also described using features extracted from the Uniprot knowledge base (`uniprot.org`).

Pairwise relations Our pairwise method is apt to make use of features that code interactions between candidate triggers and arguments. These patterns are defined from the path linking two tokens in the dependency parse tree of the sentence.

Special care was taken to perform tokenization and sentence splitting because this has an important impact on the quality of the dependency parse trees. Data was split in sentences using both the `nltk` toolkit (`nltk.org`) and the support analysis provided for the BioNLP 2013 GE task. Tokenization was carried out using a slightly modified version of the tokenizer from the Stanford event parser (McClosky et al., 2011). The dependency parse trees were finally obtained using phrase structure parser (McClosky et al., 2010)

combined with post-processing using the Stanford `corenlp` package (De Marneffe et al., 2006).

Incorporating dependency information into the pairwise model relies on the process encoding the path into feature vectors. Many formatting methods have been proposed in previous works, such as *E-walks*, that format the path into triplets (*dep*, *word*, *dep*), *V-walks* that use triplets (*word*, *dep*, *word*) or simply *N-grams* of *words*, following the dependency parse: *words* are usually encoded via stem and POS tags, and *dep* by the dependency labels (Miwa et al., 2010). All these imperfect representations lose a lot of information and can even add noise, especially when the path is long. Hence dependency parse features are processed only for pairs for which the candidate-argument path length is below a threshold whose value is a hyper-parameter.

7 Experimental Results

The hyper-parameters of our system have been optimized on the BioNLP 2013 GE task development set, after training on the corresponding training set. Using these hyper-parameter values, the final model submitted for test evaluation on the GE task server has been trained on all documents from training and development sets of BioNLP 2011 and 2013 GE tasks.

Table 3 lists the test results of our official submission. Our system achieved the best score for SIMPLE ALL and second best for PROT-MOD ALL, but it suffered from a rather poor performance on REGULATION ALL, causing a low overall score relegating the submission to the 6th place in the competition.

Event Class	recall	prec.	f-score
SIMPLE ALL	75.27	83.27	**79.07**
Binding	41.74	33.74	37.32
PROT-MOD ALL	70.68	75.84	**73.17**
REGULATION ALL	16.67	30.86	**21.64**
EVENT ALL	37.11	51.19	**43.03**

Table 3: Official test evaluation results.

After the test results were disclosed, we suspected that our poor results on REGULATION ALL were due to a bug, which was eventually discovered in the post-processing step of R-C. We re-trained our system after having fixed the bug on the latest revision of the training set (our official entry used revision 2 of the training set instead of revision 3, which resulted in slightly different annotations for *Binding* events). This led to the results displayed in Table 4 (we only show results that differ from Table 3). Our system actually achieves a EVENT ALL f-score of 51.15%, instead of 43.03%: this rating is higher than the best score of the BioNLP 2013 GE task (50.97%).

Event Class	recall	prec.	f-score
Binding	43.24	34.37	**38.30**
REGULATION ALL	31.43	47.70	**37.89**
EVENT ALL	45.96	57.66	**51.15**

Table 4: Test evaluation results after bug fix.

To compare to previous models, we also trained our system on BioNLP2011 GE task training set and evaluated it on development set. Our approach reaches a EVENT ALL f-score of 51.28%, which is lower than that of this challenge's winner, the FAUST system (Riedel et al., 2011) (55.9%). However, FAUST is a combination of several different models, compared to the UMass model (Riedel and McCallum, 2011a), which is the main constituent of FAUST, we achieve a higher EVT score (74.93% vs 74.7%) but a lower overall score (51.28% vs 54.8%). Our system is outperformed on Binding and Regulation events; this indicates the directions in which it should be improved.

8 Conclusion

This paper introduced a pairwise model designed for biomedical event extraction, which, after bug fix, outperforms the best performance of the BioNLP 2013 GE task. This system decomposes the overall task into the multi-class problem of classifying (trigger, argument) pairs. Relying of this pairwise structure for input examples allows to use joint (trigger, argument) features, to avoid costly global inference procedures over sentences and to solve a simple multi-class problem instead of a multi-label multi-class one.

Still, some issues remain. We currently cannot extract regulation events whose arguments are another regulation event. We are also subject to some cascading error between steps T-T and R-T. In future works, we intend to improve our system by turning it into a dynamic online process that will perform recursive event extraction.

Acknowledgments

This work was carried out in the framework of the Labex MS2T (ANR-11-IDEX-0004-02), and funded by the French National Agency for Research (EVEREST-12-JS02-005-01).

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.

K. Bretonnel Cohen, Karin Verspoor, Helen Johnson, Chris Roeder, Philip Ogren, William Baumgartner, Elizabeth White, and Lawrence Hunter. 2009. High-precision biological event extraction with a concept recognizer. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 50–58, Boulder, Colorado, June. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Kaibo Duan, S. Sathiya Keerthi, Wei Chu, Shirish Krishnaj Shevade, and Aun Neow Poo. 2003. Multi-category classification by soft-max combination of binary classifiers. In *In 4th International Workshop on Multiple Classifier Systems*.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. 2011. Overview of BioNLP shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA, June. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 28–36, Stroudsburg, PA, USA. Association for Computational Linguistics.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1626–1635, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

Sebastian Riedel and Andrew McCallum. 2011a. Fast and robust joint models for biomedical event extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1–12, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Sebastian Riedel and Andrew McCallum. 2011b. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 46–50, Portland, Oregon, USA, 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 BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 41–49, Boulder, Colorado, June. Association for Computational Linguistics.

Sebastian Riedel, David McClosky, Mihai Surdeanu, Andrew McCallum, and Christopher D. Manning. 2011. Model combination for event extraction in BioNLP 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 51–55, Portland, Oregon, USA, June. Association for Computational Linguistics.

Rune Sætre, Makoto Miwa, Kazuhiro Yoshida, and Jun'ichi Tsujii. 2009. From protein-protein interaction to molecular event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 103–106, Boulder, Colorado, June. Association for Computational Linguistics.

D. M. J. Tax and R. P. W. Duin. 2002. Using two-class classifiers for multiclass classification. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 124–127 vol.2.