UZH in the BioNLP 2013 GENIA Shared Task

Gerold Schneider, Simon Clematide, Tilia Ellendorff, Don Tuggener, Fabio Rinaldi,
{rinaldi,gschneid,siclemat,ellendorff,tuggener}@cl.uzh.ch
Institute of Computational Linguistics, University of Zurich, Switzerland
Gintarė Grigonytė
Stockholm University, Department of Linguistics, Section for Computational Linguistics
gintare@ling.su.se

Abstract

We describe a biological event detection method implemented for the Genia Event Extraction task of BioNLP 2013. The method relies on syntactic dependency relations provided by a general NLP pipeline, supported by statistics derived from Maximum Entropy models for candidate trigger words, for potential arguments, and for argument frames.

1 Introduction

The OntoGene team at the University of Zurich has developed text mining applications based on a combination of deep-linguistic analysis and machine learning techniques (Rinaldi et al., 2012b; Clematide and Rinaldi, 2012; Rinaldi et al., 2010). Our approaches have proven competitive in several shared task evaluations (Rinaldi et al., 2013; Clematide et al., 2011; Rinaldi et al., 2008). Additionally, we have developed advanced systems for the curation of the biomedical literature (Rinaldi et al., 2012a).

Our participation in the Genia Event Extraction task of BioNLP 2013 (Kim et al., 2013) was motivated by the desire of testing our technologies on a more linguistically motivated task. In the course of our participation we revised several modules of our document processing pipeline, however we did not have sufficient resources to completely revise the final module which generates the event structures, and we still relied on a module which we had developed for our previous participation to the BioNLP shared task.

The final submission was composed by our standard preprocessing module (described briefly in section 2) and novel probability models (section 3), combined within the old event generator (section 4).

2 Preprocessing

The OntoGene environment is based on a pipeline of several NLP tools which all operate on a common XML representation of the original document.

Briefly, the pipeline includes modules for sentence-splitting, tokenization, part-of-speech tagging, lemmatization, stemming, term-recognition (not used for the BioNLP shared task), chunking, dependency-parsing and event generation. Different variants of those modules have been used in different instantiations of the pipeline. For the BioNLP 2013 participation, lingpipe was used for sentence splitting, tokenization and PoS tagging, morpha (Minnen et al., 2001) was used for lemmatization, a python implementation of the Porter stemmer for stemming, LTTT (Grover et al., 2000), was used for chunking, and the Pro3Gres parser (Schneider, 2008) for dependency analysis.

As we have made good experiences with a rule based system for anaphora resolution in the BioNLP 2011 shared task (Tuggener et al., 2011), we implemented a similar approach that resolves anaphors to terms identified during preprocessing. Rules contain patterns like "X such as Y" or "X is a Y", and pronouns are resolved to the nearest grammatical subject or object. Anaphora resolution led to an improvement of 0.2% recall on the development set, while precision was hardly affected.

3 Probability models

Several probability models have been computed from the training data in order to be used to score and filter candidate events generated by the system. The following models played a role in the final submission:

$$P(eventType \mid trigger\ candidate) \quad (1)$$

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 116–120,
Sofia, Bulgaria, August 9 2013.

$$P(frame \wedge eventType \mid trigger\ candidate) \quad (2)$$

$$P(role \wedge eventType \mid protein) \quad (3)$$

$$P(role(t,d) \mid synpath(t,d)) \quad (4)$$

For all of them we computed global Maximum Likelihood Estimations (MLE), using the training and development datasets from the 2013 and 2011 challenges. For all of the models above, except for the last one, we also estimated the probabilities by a Maximum Entropy (ME) approach. The *MegaM* tool (Daumé III, 2004) allows for a supervised training of binary classifiers where the *class probability* is optimized by adjusting the feature weights and not just the binary *classification decision* itself. This helps to deal with the imbalanced classes such as the distribution of true or false triggerword candidates.

For the classification of *trigger candidates* (Equation 1), a binary ME classifier for each event type is separately trained, based on local and global features as described below. The triggerword candidates are collected from the training data using their stemmed representation as a selection criterion. We generally exclude triggerword candidates that occur in less than 1% as true triggers in the training set. Within the data, we found that triggers that consist of more than one word are rather rare (less than 5% of all triggers, most of them occurring once). However, we transformed these multiword triggers to singleword triggers, replacing them by their first content word.

The choice of ME features, partly inspired by (Ekbal et al., 2013), can be grouped into features derived from the triggerword itself (word), features from the sentence of the triggerword (context), and features from article-wide information (global).

Word features: (1) The text, lemma, part of speech (PoS), stem and local syntactic dependency of the triggerword candidate as computed by the Pro3Gres parser. (2) Information whether a triggerword candidate is head of a chunk as well as whether the chunk is nominal or verbal

Context features: Unigrams and bigrams in a window of variable size to left and right of the triggerword candidate; three types of uni- and bigrams are used: PoS, lemmas and stems; for unigrams we also include the lower-cased words; for bigrams, the triggerword candidate itself is included in the first bigram to either side.

Global features: (1) Presence or absence of a protein in a window of a given size around the triggerword candidate (Boolean feature); only the most frequent proteins of an article are considered. (2) The zone in an article where the triggerword candidate appears, e.g. Title/Abstract, Introduction, Background, Material and Methods, Results and Discussion, Caption and Conclusion.

Feature engineering was done by testing different combinations of settings (window size, thresholds) with the aim of finding an optimal overall ME model which reaches the lowest error rates for all event types. The error rate of the candidate set was measured as the cumulative error mass computed from the assigned class probability as follows: if the trigger candidate is a true positive, the error is 1 minus the probability assigned by the classifier. If the candidate is a false positive, the error is the probability assigned by the classifier. Our approach does not allow us to compute an error rate for false negatives, because we simply rely on the set of trigger words seen in the training data as possible candidates.

In these experiments, we discovered that for most event types an optimal setting for the context features considers a wide span of about 20 tokens to the left and right of the triggerword. Including bigrams of lemmas, stems and PoS delivered the best results compared to including only one or two of these bigram types. Context features can be parameterized according to how much positional information they contain: the distance of a word to the right and left of the trigger, only the direction (left or right) or no position information at all (bag of unigrams/bigrams). We found that the exact positional information is only important for the first word to the left and right (adjacent to the triggerword), whereas for all words that are further away it is favorable to only use the direction in relation to the trigger. A window size of 10 words within which proteins are found in the context of a triggerword gave the best results. The optimal number of the most frequent proteins considered within this window was found to be the 10 most frequent proteins within an article.

The second type of ME classifier (Equation 2) has the purpose of calculating the probabilities of event frames for all event types given a trigger word. We use the term *frame* for a combination of arguments that an event is able to accept as theme and cause and whether these arguments are real-

ized as proteins or subevents.

For the classification of *proteins* (Equation 3), again separate binary ME classifiers were built in order to estimate the probability that a protein has a role (theme or cause) in an event of a given type.

4 Event Generation

We tested two independent event generation modules, one based on a revision of our previous 2009 submission (Kaljurand et al., 2009) and one which is a totally new implementation. We could do only preliminary tests with the second module, which however showed promising results, in particular with much better recall than the older module (up to 65.23%), despite the very little time that we could invest in its development. The best F-score that we could reach was still slightly inferior to the one of the old module at the deadline for submission of results. In the rest of this paper we will describe only the module which was used in the official submission.

The event extraction process consists of three phases. First, event candidates are generated, based on trigger words and their context, using the ME and MLE probabilities p_T (equation 1).

Second, individual arguments of an event are generated. We calculate the MLE probability p_R of an argument role (e.g. Theme) to occur as part of a given event type, as follows:

$$p_R(Role\,|EventType) = \frac{f(Role \wedge EventType)}{f(EventType)} \quad (5)$$

We obtained the best results on the development corpus when combining the probabilities as:

$$p_A = \frac{p_T * p_T * p_R}{p_T + p_T + p_R} \quad (6)$$

We generate arguments, using an MLE syntactic path and an ME argument model, as follows. The syntactic path between the trigger word and every term (protein or subordinate event) is considered. If they are syntactically connected, and if the probability of a syntactic path to express an event is above a threshold, it is selected. As this is a filtering step, it negatively affects recall.

We calculate the MLE probability p_{path} that a syntactic configuration fills an argument slot. Syntactic configurations consist of the head word (trigger) *HWord*, the head event type *HType*, the dependent word *DWord*, the dependent event type *DType*, and the syntactic path *Path* between them.

In order to deal with sparse data, we use a smoothed model.

$$\begin{aligned} p_{path}(Arg\,|\,HWord,\,HType,\,DWord,\,DType,\,Path) \;=\; & \\ \tfrac{1}{w_1+w_2+w_3} * (& \\ w_1 * \tfrac{f(HWord,\,HType,\,DWord,\,DType,\,Path \wedge Arg)}{f(HWord,\,HType,\,DWord,\,DType,\,Path)} \;+ & \\ w_2 * \tfrac{f(HType,\,DType,\,Path \wedge Arg)}{f(HType,\,DType,\,Path)} \;+ & \\ w_3 * \tfrac{f(HType,\,DType \wedge Arg)}{f(HType,\,DType)}) & \quad (7) \end{aligned}$$

The weights were emprically set as $w_1 = 4$, $w_2 = 2$ and $w_3 = 1.5$. The fact that the weights decrease approximates a back-off model. The final probability had to be larger than 0.2.

We have also used an ME model which delivers the probability p_{arg} that a term is the argument of a specific event, see formula 3. If this ME model predicts with a probability of above 80% that the term is not an argument, the search fails. Otherwise, the probabilities are combined. On the development corpus, we achieved best results when using the harmonic mean:

$$p_{argument} = 2 * \frac{p_{path} * p_{arg}}{p_{path} + p_{arg}} \quad (8)$$

As a last step, the several arguments of an event are combined into a frame. We have tested models predicting an entire frame directly, and models combining the individual arguments generated in the previous step. The latter approach performed better. Any permutation of the argument candidates could constitute a frame. Only frames seen in the training corpus for a given event type are considered. We have again used an ME and an MLE model for predicting frames.

The ME model predicts $p_{frame.ME}$, see formula 2. We have also used two MLE models: the first one delivers the probability $p_{frame.MLE}$ based on the event type only, the second one $p_{frameword.MLE}$ also considers the trigger word and is much sparser (a low default is thus used for unseen words). The probability of the individual arguments also needs to be taken into consideration. We used the mean of the individual arguments' probabilities ($p_{args'mean}$).

5 Evaluation

In our analysis of errors, we noticed that frames with more than one argument are created extremely rarely. The problem is that frames with several arguments are rarer because the context often does not offer the possibility to attach several arguments. Therefore, we consistently undergenerated with $p_{args'mean}$ as outlined above.

Event Class	gold (match)	answer (match)	recall	prec.	fscore
SVT-TOTAL	1117 (619)	851 (619)	55.42	72.74	**62.91**
EVT-TOTAL	1490 (698)	1103 (698)	46.85	63.28	**53.84**
REG-TOTAL	1694 (168)	618 (168)	9.92	27.18	**14.53**
All events total	3184 (866)	1721 (866)	27.20	50.32	**35.31**

Table 1: Results on the development set, measured using "strict equality".

Event Class	gold (match)	answer (match)	recall	prec.	fscore
Gene_expression	619 (400)	497 (400)	64.62	80.48	71.68
Transcription	101 (26)	100 (26)	25.74	26.00	25.87
Protein_catabolism	14 (10)	15 (10)	71.43	66.67	68.97
Localization	99 (34)	39 (34)	34.34	87.18	49.28
=[SIMPLE ALL]=	833 (470)	651 (470)	56.42	72.20	63.34
Binding	333 (74)	264 (74)	22.22	28.03	24.79
Protein_modification	1 (0)	0 (0)	0.00	0.00	0.00
Phosphorylation	160 (119)	168 (119)	74.38	70.83	72.56
Ubiquitination	30 (0)	0 (0)	0.00	0.00	0.00
Acetylation	0 (0)	0 (0)	0.00	0.00	0.00
Deacetylation	0 (0)	0 (0)	0.00	0.00	0.00
=[PROT-MOD ALL]=	191 (119)	168 (119)	62.30	70.83	66.30
Regulation	288 (23)	84 (23)	7.99	27.38	12.37
Positive_regulation	1130 (129)	444 (129)	11.42	29.05	16.39
Negative_regulation	526 (54)	166 (54)	10.27	32.53	15.61
=[REGULATION ALL]=	1944 (206)	694 (206)	10.60	29.68	15.62
==[EVENT TOTAL]==	3301 (869)	1777 (869)	26.33	48.90	34.23

Table 2: Results on the test data, measured using "strict equality".

We have added a number of heuristics to boost multi-argument frames. Multiplying the probability of a frame by its cubed length (giving two-argument slots 9 times higher probability), and giving Cause-slots 50% higher scores globally led to best results.

We mainly trained and evaluated using the "strict equality" evaluation criteria as our reference. The results on the development data are shown in table 1. With more relaxed equality definitions, the results were always a few percentage points better. Our results in the official test run are shown in table 2. In sum, our submitted system has good performance for simple events, bad performance for *Binding* events, and a bias towards precision due to a syntactic-based filtering step.

6 Conclusions and Future work

Our participation in the 2013 BioNLP shared task was a useful opportunity to revise components of the OntoGene pipeline and begin the implementation of a novel event generator. Due to lack of time, it was not completed in time for the official submission. We will continue its development and use the BioNLP datasets.

Acknowledgments

This research is partially funded by the Swiss National Science Foundation (grant 105315_130558/1).

References

[Clematide and Rinaldi2012] Simon Clematide and Fabio Rinaldi. 2012. Ranking relations between diseases, drugs and genes for a curation task. *Journal of Biomedical Semantics*, 3(Suppl 3):S5.

[Clematide et al.2011] Simon Clematide, Fabio Rinaldi, and Gerold Schneider. 2011. Ontogene at calbc ii and some thoughts on the need of document-wide harmonization. In *Proceedings of the CALBC II workshop, EBI, Cambridge, UK, 16-18 March*.

[Daumé III2004] Hal Daumé III. 2004. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at `http://pub.hal3.name#daume04cg-bfgs`, implementation available at `http://hal3.name/megam/`, August.

[Ekbal et al.2013] Asif Ekbal, Sriparna Saha, and Sachin Girdhar. 2013. Evolutionary approach for classifier ensemble: An application to bio-molecular event extraction. In Ajith Abraham and Sabu M Thampi, editors, *Intelligent Informatics*, volume 182 of *Advances in Intelligent Systems and Computing*, pages 9–15. Springer Berlin Heidelberg.

[Grover et al.2000] Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. 2000. Lt ttt - a flexible tokenisation tool. In *Proceedings of Second International Conference on Language Resources and Evaluation (LREC 2000)*.

[Kaljurand et al.2009] Kaarel Kaljurand, Gerold Schneider, and Fabio Rinaldi. 2009. UZurich in the BioNLP 2009 Shared Task. In *Proceedings of the BioNLP workshop, Boulder, Colorado*.

[Kim et al.2013] Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. 2013. The genia event extraction shared task, 2013 edition - overview. In *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Minnen et al.2001] Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

[Rinaldi et al.2008] Fabio Rinaldi, Thomas Kappeler, Kaarel Kaljurand, Gerold Schneider, Manfred Klenner, Simon Clematide, Michael Hess, Jean-Marc von Allmen, Pierre Parisot, Martin Romacker, and Therese Vachon. 2008. OntoGene in BioCreative II. *Genome Biology*, 9(Suppl 2):S13.

[Rinaldi et al.2010] Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Simon Clematide, Therese Vachon, and Martin Romacker. 2010. OntoGene in BioCreative II.5. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):472–480.

[Rinaldi et al.2012a] Fabio Rinaldi, Simon Clematide, Yael Garten, Michelle Whirl-Carrillo, Li Gong, Joan M. Hebert, Katrin Sangkuhl, Caroline F. Thorn, Teri E. Klein, and Russ B. Altman. 2012a. Using ODIN for a PharmGKB re-validation experiment. *Database: The Journal of Biological Databases and Curation*.

[Rinaldi et al.2012b] Fabio Rinaldi, Gerold Schneider, and Simon Clematide. 2012b. Relation mining experiments in the pharmacogenomics domain. *Journal of Biomedical Informatics*, 45(5):851–861.

[Rinaldi et al.2013] Fabio Rinaldi, Simon Clematide, Simon Hafner, Gerold Schneider, Gintare Grigonyte, Martin Romacker, and Therese Vachon. 2013. Using the ontogene pipeline for the triage task of biocreative 2012. *The Journal of Biological Databases and Curation, Oxford Journals*.

[Schneider2008] Gerold Schneider. 2008. *Hybrid Long-Distance Functional Dependency Parsing*. Doctoral Thesis, Institute of Computational Linguistics, University of Zurich.

[Tuggener et al.2011] D Tuggener, M Klenner, G Schneider, S Clematide, and F Rinaldi. 2011. An incremental model for the coreference resolution task of bionlp 2011. In *BioNLP 2011*, pages 151–152. Association for Computational Linguistics (ACL), June.