NaCTeM EventMine for BioNLP 2013 CG and PC tasks

Makoto Miwa and **Sophia Ananiadou**
National Centre for Text Mining, University of Manchester, United Kingdom
School of Computer Science, University of Manchester, United Kingdom
{makoto.miwa,sophia.ananiadou}@manchester.ac.uk

Abstract

This paper describes NaCTeM entries for the Cancer Genetics (CG) and Pathway Curation (PC) tasks in the BioNLP Shared Task 2013. We have applied a state-of-the-art event extraction system EventMine to the tasks in two different settings: a single-corpus setting for the CG task and a stacking setting for the PC task. EventMine was applicable to the two tasks with simple task specific configuration, and it produced a reasonably high performance, positioning second in the CG task and first in the PC task.

1 Introduction

With recent progress in biomedical natural language processing (BioNLP), automatic extraction of biomedical events from texts becomes practical and the extracted events have been successfully employed in several applications, such as EVEX (Björne et al., 2012; Van Landeghem et al., 2013) and PathText (Miwa et al., 2013a). The practical applications reveal a problem in that both event types and structures need to be covered more widely. The BioNLP Shared Task 2013 (BioNLP-ST 2013) offers several tasks addressing the problem, and especially in the Cancer Genetics (CG) (Pyysalo et al., 2013) and Pathway Curation (PC) (Ohta et al., 2013) tasks, new entity/event types and biomedical problems are focused.

Among dozens of extraction systems proposed during and after the two previous BioNLP shared tasks (Kim et al., 2011; Kim et al., 2012; Pyysalo et al., 2012b), EventMine (Miwa et al., 2012)[1] has been applied to several biomedical event extraction corpora, and it achieved the state-of-the-art performance in several corpora (Miwa et al., 2013b). In these tasks, an event associates with a trigger expression that denotes its occurrence in text, has zero or more arguments (entities or other events) that are identified with their roles (e.g., *Theme*, *Cause*) and may be assigned hedge attributes (e.g., *Negation*).

This paper describes how EventMine was applied to the CG and PC tasks in the BioNLP-ST 2013. We configured EventMine minimally for the CG task and submit the results using the models trained on the training and development data sets with no external resources. We employed a stacking method for the PC task; the method basically trained the models on the training and development data sets, but it also employed features representing prediction scores of models on seven external corpora.

We will first briefly describe EventMine and its task specific configuration in the next section, then show and discuss the results, and finally conclude the paper with future work.

2 EventMine for CG and PC Tasks

This section briefly introduces EventMine and the PC and CG tasks, and then explains its task specific configuration.

2.1 EventMine

EventMine (Miwa et al., 2012) is an SVM-based pipeline event extraction system. For the details, we refer the readers to Miwa et al. (2012; 2013b). EventMine consists of four modules: a trigger/entity detector, an argument detector, a multi-argument detector and a hedge detector. The trigger/entity detector finds words that match the head words (in their surfaces, base forms by parsers, or stems by a stemmer) of triggers/entities in the training data, and the detector classifies each word into specific entity types (e.g., *DNA_domain_or_region*), event types (*Regulation*) or a negative type that represents the word does not participate in any events. The argument

[1] http://www.nactem.ac.uk/EventMine/

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 94–98,
Sofia, Bulgaria, August 9 2013.

detector enumerates all possible pairs among triggers and arguments that match the semantic type combinations of the pairs in the training data, and classifies each pair into specific role types (e.g., *Binding:Theme-Gene_or_gene_product*) or a negative type. Similarly, the multi-argument detector enumerates all possible combinations of pairs that match the semantic type structures of the events in the training data, and classifies each combination into an event structure type (e.g., *Positive_regulation:Cause-Gene_or_gene_product:Theme-Phosphorylation*) or a negative type. The hedge detector attaches hedges to the detected events by classifying the events into specific hedge types (*Speculation* and *Negation*) or a negative type.

All the classifications are performed by one-vs-rest support vector machines (SVMs). The detectors use the types mentioned above as their classification labels. Labels with scores larger than the separating hyper-plane of SVM and the label with the largest value are selected as the predicted labels; the classification problems are treated as multi-class multi-label classification problems and at least one label (including a negative type) needs to be selected in the prediction.

Features for the classifications include character n-grams, word n-grams, shortest paths among event participants on parse trees, and word n-grams and shortest paths between event participants and triggers/entities outside of the events on parse trees. The last features are employed to capture the dependencies between the instances. All gold entity names are replaced with their types, the feature space is compressed to 2^{20} by hashing to reduce space cost, the positive instances are weighted to reduce class imbalance problems, the feature vectors are normalised, and the C parameter for SVM is set to 1.

In the pipeline approach, there is no way to detect instances if the participants are missed by the preceding modules. EventMine thus aims high recall in the modules by the multi-label setting and weighting positive instances. EventMine also avoids training on instances that cannot be detected by generating the training instances based on predictions by the preceding modules since the training and test instances should be similar.

EventMine is flexible and applicable to several event extraction tasks with task specific configuration on entity, role and event types. This configuration is described in a separate file[2].

2.2 CG and PC Tasks

The CG task (Pyysalo et al., 2013) aims to extract information on the biological processes relating to the development and progression of cancer. The annotation is built on the Multi-Level Event Extraction (MLEE) corpus (Pyysalo et al., 2012a), which EventMine was once applied to. The PC task (Ohta et al., 2013), on the other hand, aims to support the curation of bio-molecular pathway models, and the corpus texts are selected to cover both signalling and metabolic pathways.

Both CG and PC tasks offer more entity, role and event types than most previous tasks like GENIA (Kim et al., 2012) does, which may make the classification problems more difficult.

2.3 Configuration for CG and PC Tasks

We train models for the CG and PC tasks in similar configuration, except for the incorporation of a stacking method for the PC task. We first explain the configuration applied to both tasks and then introduce the stacking method for the PC task.

We employ two kinds of type generalisations for both tasks: one for the classification labels and features and the other for the generation of instances. After the disambiguation of trigger/entity types by the trigger/entity detector, we reduce the number of event role labels and event structure labels by the former type generalisations. The generalisations are required to reduce the computational costs that depend on the number of the classification labels. Unfortunately, we cannot evaluate the effect of the generalisations on the performance since there are too many possible labels in the tasks. The generalisations may alleviate the data sparseness problem but they may also induce over-generalised features for the problems with enough training instances. For event roles, we generalise regulation types (e.g., *Positive_regulation*, *Regulation*) into a single *REGULATION* type and post-transcriptional modification (PTM) types (e.g., *Acetylation*, *Phosphorylation*) into a single *PTM* type for trigger types, numbered role types into a non-numbered role type (e.g., *Participant2*→*Participant*) for role

[2]This file is not necessary since the BioNLP ST data format defines where these semantic types are described, but this file is separated for the type generalisations explained later and the specification of gold triggers/entities without reproducing a1/a2 files.

types, and event types into a single *EVENT* type and entity types into a single *ENTITY* type for argument types. For event structures, we apply the same generalisations except for the generalisations of numbered role types since the numbered role types are important in differentiating events. Unlike other types, the numbered role types in events are not disambiguated by any other modules. The generalisations are also applied to the features in all the detectors when applicable. These generalisations are the combination of the generalisations for the GENIA, Epigenetics and Post-translational Modifications (EPI), and Infectious Diseases (ID) (Pyysalo et al., 2012b) of the BioNLP-ST 2011 (Miwa et al., 2012).

The type generalisations on labels and features are not directly applicable to generate possible instances in the detectors since the generalisations may introduce illegal or unrealistic event structures. Instead, we employ separate type generalisations to expand the possible event role pair and event structure types and cover types, which do not appear in the training data. For example, if there are *Regulation:Theme-Gene_expression* instances but there are no *Positive_regulation:Theme-Gene_expression* instances in the training data, we allow the creation of the latter instances by generalising the triggers, i.e., *REGULATION:Theme-Gene_expression*, and we used all the created instances for classification. The type generalisations may incorporate noisy instances but they pose the possibility to find unannotated event structures. To avoid introducing unexpected event structures, we apply the generalisations only to the regulation trigger types.

We basically follow the setting for EPI in Miwa et al. (2012). We employ a deep syntactic parser Enju (Miyao and Tsujii, 2008) and a dependency parser GDep (Sagae and Tsujii, 2007). We utilise liblinear-java (Fan et al., 2008)[3] with the L2-regularised L2-loss linear SVM setting for the SVM implementation, and Snowball[4] for the stemmer. We, however, use no external resources (e.g., dictionaries) or tools (e.g., a coreference resolver) except for the external corpora in the stacked models for the PC task.

We train models for the CG task using the configuration described above. For PC, in addition to the configuration, we incorporated a stacking method (Wolpert, 1992) using the models with the same configuration for seven other available corpora: GENIA, EPI, ID, DNA methylation (Ohta et al., 2011a), Exhaustive PTM (Pyysalo et al., 2011), mTOR (Ohta et al., 2011b) and CG. The prediction scores of all the models are used as additional features in the detectors. Although some corpora may not directly relate to the PC task and models trained on such corpora can produce noisy features, we use all the corpora without selection since the stacking often improve the performance, e.g., (Pyysalo et al., 2012a; Miwa et al., 2013b).

Setting	Recall	Precision	F-score
–	42.87	47.72	45.16
+Exp.	43.37	46.42	44.84
+Exp.+Stack.	43.59	48.77	46.04

Table 1: Effect of the type generalisations for expanding possible instances (+Exp.) and stacking method (+Stack.) on the PC development data set.

3 Evaluation

We first evaluate the type generalisations for expanding possible event structures and the stacking method in Table 1. The scores were calculated using the evaluation script provided by the organisers with the official evaluation metrics (soft boundary and partial recursive matching). The generalisations improved recall with the loss of precision, and they slightly degraded the F-score in total. The generalisations were applied to the test set in the submission since this result was expected as explained in Section 2.3 and the slightly high recall is favourable for the practical applications like semantic search engines (Miwa et al., 2013a). Although the improvement by the stacking method (+Exp.+Stack. compared to +Exp.) is not statistically significant (p=0.14) using the approximate randomisation method (Noreen, 1989; Kim et al., 2011), this slight improvement indicates that the corpus in the PC task shares some information with the other corpora.

Tables 2 and 3 show the official scores of our entries on the test data sets for the CG and PC tasks[5]. EventMine ranked second in the CG task and first in the PC task. The scores of the best system among the other systems (TEES-2.1 (Björne and Salakoski, 2013)) are shown for reference.

[3] http://liblinear.bwaldvogel.de/

[4] http://snowball.tartarus.org/

[5] We refer to the websites of the tasks for the details of the event categories.

Task	System	Rec.	Prec.	F-Score
CG	EventMine	**48.83**	55.82	52.09
	TEES-2.1	48.76	**64.17**	**55.41**
PC	EventMine	**52.23**	53.48	**52.84**
	TEES-2.1	47.15	**55.78**	51.10

Table 2: Official best and second best scores on the CG and PC tasks. Higher scores are shown in bold.

Task	Category	EventMine	TEES-2.1
CG	ANATOMY	71.31	**77.20**
	PATHOL	59.78	**67.51**
	MOLECUL	**72.77**	72.60
	GENERAL	**53.08**	52.20
	REGULAT	39.79	**43.08**
	PLANNED	**40.51**	39.43
	MOD	29.95	**34.66**
PC	SIMPLE	**65.60**	63.92
	NON-REG	**65.72**	63.37
	REGULAT	**40.10**	39.39
	MOD	28.05	**28.73**

Table 3: F-scores on the CG and PC tasks for event categories. Higher scores are shown in bold.

EventMine achieved the highest recall for both tasks, and this is favourable as mentioned above. This high recall is reasonable since EventMine solved the problems as multi-label classification tasks, corrected the class imbalance problem as explained in Section 2.1 and incorporated the type generalisations for expanding possible event structures. The performance (in F-score) on both CG and PC tasks is slightly lower than the performance on the GENIA and ID tasks in the BioNLP-ST 2011 (Miwa et al., 2012), and close to the performance on the EPI task. This may be partly because the GENIA and ID tasks deal with a fewer number of event types than the other tasks.

EventMine performed worse than the best system in the CG task, but this result is promising considering that we did not incorporate any other resources and tune the parameters (e.g., C in SVM). The detailed comparison with TEES-2.1 shows that EventMine performed much worse than TEES-2.1 in anatomical and pathological event categories, which contained relatively new event types. This indicates EventMine missed some of the new structures in the new event types.

The range of the scores is similar to the scores on the MLEE corpus (52.34–53.43% in F-Score (Pyysalo et al., 2012a)) although we cannot directly compare the results. The ranges of the scores are around 60% to 70% for non-nested events (e.g., *SIMPLE*), 40% for nested events (e.g., *REGULAT*) and 30% for modifications (e.g., *MOD*). This large spread of the scores may be caused by a multiplication of errors in predicting their participants, since similar spread was seen in the previous tasks (e.g., (Miwa et al., 2012)). These results indicate that we may not be able to improve the performance just by increasing the training instances.

These results show that EventMine performed well on the PC task that is a completely novel task for EventMine, and the stacking would also work effectively on the test set.

4 Conclusions

This paper explained how EventMine was applied to the CG and PC tasks in the BioNLP-ST 2013. EventMine performed well on these tasks and achieved the second best performance in the CG task and the best performance in the PC task. We show the usefulness of incorporating other existing corpora in the PC task. The success of this application shows that the EventMine implementation is flexible enough to treat the new tasks. The performance ranges, however, shows that we may need to incorporate other novel techniques/linguistic information to produce the higher performance.

As future work, we will investigate the cause of the missed events. We also would like to extend and apply other functions in EventMine, such as co-reference resolution, and seek a general approach that can improve the event extraction performance on all the existing corpora, using the training data along with external resources.

Acknowledgement

This work is supported by the Biotechnology and Biological Sciences Research Council (BBSRC) [BB/G53025X/1] and the Grant-in-Aid for Young Scientists B [25730129] of the Japan Science and Technology Agency (JST).

References

Jari Björne and Tapio Salakoski. 2013. TEES 2.1: Automated annotation scheme learning in the bioNLP

2013 shared task. In *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jari Björne, Sofie Van Landeghem, Sampo Pyysalo, Tomoko Ohta, Filip Ginter, Yves Van de Peer, Sophia Ananiadou, and Tapio Salakoski. 2012. Pubmed-scale event extraction for post-translational modifications, epigenetics and protein structural relations. In *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 82–90, Montréal, Canada, June. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2011. Extracting Bio-Molecular Events from Literature – the BioNLP'09 Shared Task. *Computational Intelligence*, 27(4):513–540.

Jin-Dong Kim, Ngan Nguyen, Yue Wang, Jun'ichi Tsujii, Toshihisa Takagi, and Akinori Yonezawa. 2012. The Genia Event and Protein Coreference tasks of the BioNLP Shared Task 2011. *BMC Bioinformatics*, 13(Suppl 11):S1.

Makoto Miwa, Paul Thompson, and Sophia Ananiadou. 2012. Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. *Bioinformatics*, 28(13):1759–1765.

Makoto Miwa, Tomoko Ohta, Rafal Rak, Andrew Rowley, Douglas B. Kell, Sampo Pyysalo, and Sophia Ananiadou. 2013a. A method for integrating and ranking the evidence for biochemical pathways by mining reactions from text. *Bioinformatics*. (In Press).

Makoto Miwa, Sampo Pyysalo, Tomoko Ohta, and Sophia Ananiadou. 2013b. Wide coverage biomedical event extraction using multiple partially overlapping corpora. *BMC Bioinformatics*, 14(1):175.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80, March.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses : An Introduction*. Wiley-Interscience, April.

Tomoko Ohta, Sampo Pyysalo, Makoto Miwa, and Jun'ichi Tsujii. 2011a. Event extraction for dna methylation. *Journal of Biomedical Semantics*, 2(Suppl 5):S2.

Tomoko Ohta, Sampo Pyysalo, and Jun'ichi Tsujii. 2011b. From pathways to biomolecular events: Opportunities and challenges. In *Proceedings of BioNLP'11*, pages 105–113, Portland, Oregon, USA. ACL.

Tomoko Ohta, Sampo Pyysalo, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, and Sophia Ananiadou. 2013. Overview of the pathway curation (PC) task of bioNLP shared task 2013. In *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, and Jun'ichi Tsujii. 2011. Towards exhaustive event extraction for protein modifications. In *Proceedings of BioNLP'11*, pages 114–123, Portland, Oregon, USA, June. ACL.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun'ichi Tsujii, and Sophia Ananiadou. 2012a. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581.

Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. 2012b. Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011. *BMC Bioinformatics*, 13(Suppl 11):S2.

Sampo Pyysalo, Tomoko Ohta, and Sophia Ananiadou. 2013. Overview of the cancer genetics (CG) task of bioNLP shared task 2013. In *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050, Prague, Czech Republic, June. ACL.

S. Van Landeghem, J. Bjorne, C. H. Wei, K. Hakala, S. Pyysalo, S. Ananiadou, H. Y. Kao, Z. Lu, T. Salakoski, Y. Van de Peer, and F. Ginter. 2013. Large-scale event extraction from literature with multi-level gene normalization. *PLoS One*, 8(4):e55814.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.