Event Extraction in pieces: Tackling the partial event identification problem on unseen corpora

Chrysoula Zerva
National Center of Text Mining,
School of Computer Science,
University of Manchester, UK
c.zerva@cs.man.ac.uk

Sophia Ananiadou
National Center of Text Mining,
School of Computer Science,
University of Manchester, UK
sophia.ananiadou@manchester.ac.uk

Abstract

Biomedical event extraction systems have the potential to provide a reliable means of enhancing knowledge resources and mining the scientific literature. However, to achieve this goal, it is necessary that current event extraction models are improved, such that they can be applied confidently to unseen data with a minimal rate of error. Motivated by this requirement, this work targets a particular type of error, namely partial events, where an event is missing one or more arguments. Specifically, we attempt to improve the performance of a state-of-the-art event extraction tool, EventMine, when applied to a new cancer pathway curation corpus. We propose a post-processing ranking approach based on relaxed constraints, in order to reconsider the candidate arguments for each event trigger, and suggest possible new arguments. The proposed methodology, applicable to the output of any event extraction system, achieves an improvement in argument recall of 2%-4% when applied to EventMine output, and thus constitutes a promising direction for further developments.

1 Introduction

In text mining, events are currently the most complex information unit that can be extracted from raw text, in terms of their ability to capture n-ary dynamic relations between entities and/or other events as indicated in Figure 1. Their dynamic properties mean that events constitute the closest equivalent to human-extracted information. The structured information representation of events can be used to enrich current knowledge sources such as ontologies and databases in an automated manner. This can be particularly useful for researchers in the biomedical domain, who use complicated models to represent molecular reactions, pathways etc. In order to improve these models, biologists currently need to sift through a continuously growing mountain of literature (Ananiadou et al., 2014). Thus, an automated means to extract knowledge, using event extraction technology, and to exploit this knowledge to augment existing models, would be an immense asset within biomedical research.

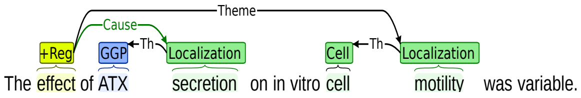
Theme
Cause
Th
Th
+Reg
GGP
Localization
Cell
Localization
The effect of ATX secretion on in vitro cell motility was variable.

Figure 1: Event Extraction example: Localisation events nested as arguments to Positive Regulation (+Reg) event [1]

Motivated by the above, the Big Mechanism project (Cohen, 2014) aims to augment cancer pathway models automatically with events extracted from biomedical literature. To this end, event extraction systems need to be able not only to extract high quality events that cover a wide range of biomedical event types but also to robustly do so even when applied to unseen data. Indeed, the expectation is that event extraction systems will be successful in carrying out this task even when parameters such as text type or domain, are altered, without the need to retain the system.

However, the structure of current event extraction systems can hinder the ability to achieve the above goal. Since event extraction has so far been treated as a supervised learning task, the performance of systems is heavily dependent on the annotation, context and domain of the training data, and may drop significantly when one of the initial

[1] Sentence taken BioNLP 2013 CG corpus (Nédellec et al., 2013). Annotation visualised with BRAT annotation tool (Stenetorp et al., 2012).

Proceedings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP 2015), pages 31–41,
Beijing, China, July 30, 2015. ©2015 Association for Computational Linguistics

specifications changes, even within the same domain. Especially in pipelined architectures, which consist of sequential classification tasks, additional annotation constraints are learned from the training corpus at each stage in the pipeline. While these additional constraints improve the model's precision, they render it less adaptable to deviating event structures. One of the consequences of this is the failure to retrieve some of the information that should be associated with the event, leading to so-called partial event identification, where some of the event arguments are missing. Although such errors may not be of vital importance in all domains, they can be extremely detrimental when attempting to link an event to a biomedical model, since they can lead to erroneous or useless assertions.

The work described here focusses on resolving the problem by applying a generic constraint relaxation post-processing strategy to the output of an event extraction system (EventMine (Miwa and Ananiadou, 2013)), with the aim of reducing the number of recognised events that have missing arguments. Motivated by an analysis of the Big Mechanism testing corpus described in Section 3, we relax the annotation constraints related to argument roles and subsequently reconsider all the entities within a sentence that are valid argument candidates, by exploiting syntactical dependencies. We employ the confidence values obtained from an *Adaboost* (Freund and Schapire, 1997) classifier to rank candidate arguments for each event trigger, and to determine which of them constitute valid additions to the event. Using this approach, we are able to improve the *recall* on partial events identified by EventMine by at least 2% and, importantly, we gain fruitful insights into factors that could further improve performance.

2 Background: The Event Extraction Task in Biomedicine

In this section we provide an overview of the event extraction procedure, focussing on biomedical events. Our emphasis is on the details of pipelined event extraction, since this is the approach employed by the EventMine system, which we use to perform event extraction. Finally, we review the main approaches for adapting event extraction to new or unseen data.

2.1 Event Structure

In text mining, events refer to units representing dynamic, n-ary relations between named entities. In the biomedical domain, this definition can be narrowed to units representing molecular interactions stated within textual documents (Björne and Salakoski, 2011).

The typical structure of events (as defined and used in BioNLP shared tasks, e.g., (Kim et al., 2009), (Nédellec et al., 2013)) includes an obligatory *predicate/trigger*, i.e., a word sequence in text that characterises the event type. Potentially, an event may also have one or more *arguments*, i.e., entities in text that are semantically linked to the trigger. Considering the trigger and the arguments as nodes, the links between them can be considered as directed edges (from the trigger to the argument), which represent the role that the argument plays with respect to the trigger. As events are dynamic elements, the same entity can participate in different events, and may assume different roles in each event. Also, since events are solid information units, they can themselves act as arguments to other events, leading to the extraction of complex/nested events (Björne et al., 2010). These characteristics can be observed in the example of Fig 1 presented in Section 1.

2.2 Event Mining Architecture

In order to extract structures of the complexity illustrated in Figure 1, current state-of-the-art systems break event extraction down into multiple classification tasks that have to be solved in order to produce the final structured event representation. The learning process to carry out these tasks can be undertaken either sequentially in a pipelined manner, as in EventMine (Miwa and Ananiadou, 2013) and TEES (Björne and Salakoski, 2013), or as joint learning task, as for FAUST (Riedel and McCallum, 2011). EventMine, the system employed in this work, utilises the pipelined approach, and consists of the following modules:

- *Event trigger classifier:* Identifies spans of text that act as triggers and annotates them with the corresponding type (event label).
- *Argument detector:* Links each trigger with at most one argument and annotates the edge (link) with the corresponding argument role type.
- *Multiple argument detector:* Adds additional arguments to the pairs of the previous step, final-

ising each event structure.

- *Modification detector:* Identifies event modifications (negation and speculation)

All the above are formulated as multi-class tasks that are learned in a supervised way, using one-versus-rest SVM implementation of LibLinear (Fan et al., 2008). EventMine is able to perform with state-of-the-art accuracy, achieving F-score of 52% on the CG and 53% on PC task of the latest BioNLP shared task (Nédellec et al., 2013), rendering it a suitable tool for this study.

2.3 Adaptation and generalisation approaches

One of the problems usually encountered with supervised models, such as those used by EventMine, is that they are specifically tailored to features of the corpus on which they have been trained. As a result, their functionality is restricted to the trigger, argument and role types that they have been trained to identify and extract. For example, some corpora focus only on protein-protein interactions, while others include chemical reactions, anatomical entities and or a combination of the above. Intuitively, in order to capture events that encompass all the above types, either one would have to re-annotate a corpus with all the required types of interest, or use some combination of either the corpora or the models trained on them.

Since corpus annotation is an expensive and time consuming task, various computational approaches to combining information have been proposed. A particularly straightforward approach is to combine the models in a stacking manner as in (Wolpert, 1992), where a method inspired from cross-validation is used to train different models on subsets of the different corpora, and then use the validation set to learn how to combine their outputs to obtain the desired result. More recently, a range of domain adaptation techniques have been proposed that try to adapt to a new corpus by either selectively training on the instances and/or features that are expected to maximize performance (Chen et al., 2011; Xia et al., 2013), or by attempting to tailor feature distributions to the one of the new corpus with various methods such as kernel based ones (Daumé III, 2009; Kulis et al., 2011) or transfer component analysis (Pan et al., 2011). Finally, (Miwa et al., 2013) suggests the use of a filtering model, which considers the overlap of the available corpora and filters redundant and contradicting labelling across different corpora and then merges the corpora in order to train a single model on their combination. The filtering, as Miwa explains, is heuristically achieved by limiting the generation of negative instances in each corpus to only those cases in which the corresponding surface expression matches at least one positive instance of an annotated type in any corpus that shares that type. The method, referred to as wide coverage, when implemented in EventMine outperforms other stacking and domain adaptation methods as shown in (Miwa et al., 2013). Accordingly, it was the chosen approach for this work.

3 Corpora and Annotation Considerations

3.1 Training Corpora

For training the wide coverage method was applied on the combination of the training sets of the following corpora, treated as described in (Miwa et al., 2013) : Genia09 of BioNLP '09 (Kim et al., 2009), Genia11, EPI & ID of BioNLP '11 (Kim et al., 2011), DNA-methylation (Ohta et al., 2011), ePTM (Pyysalo et al., 2011), mTOR (Caron et al., 2010), and MLEE (Pyysalo et al., 2012).

3.2 Testing Corpora

The corpus that provided the motivation for this work, henceforth referred to as BM, is a small annotated set of six passages extracted from full-text biomedical research papers in PubMed [2]. It concerns cancer pathway curation and was manually annotated with biomedical named entities and events by expert biologists participating in the Big Mechanism project (Cohen, 2014). In total it consists of 155 event and 247 named entity annotations. The range of the entities and events annotated render it a valid candidate for the application of the wide coverage approach described in Section 2.3, because the entities span across *Chemical, Protein* and *Cell* instances, while the event types cover pathways, various protein interactions (*Binding, Regulation*, etc) and other cancer related events. Since there is no single related training corpus with similar annotations, a model that can learn labels from different corpora is necessary to facilitate recognition of all of the above event and entity types.

[2] PubMed ids: PMC2872605, PMC3058384

The annotation scheme in BM corpus differs from the uniform annotation scheme used in the training corpora in the following ways:

- The entity labels are different to those used in the training corpora (see Fig 3 and section 4.1)
- No distinction is made between different event types in the test corpus annotations
- A simplified edge type annotation was followed in the BM corpus, discriminating only between simple arguments and arguments indicating the site (cellular location) where the event took place. As opposed to the BioNLP schema the edge annotations in BM contain less semantic information (there is no discrimination between roles such as Instrument, Participant, Cause etc)

Figure 2 illustrates an event annotated according to both BioNLP guidelines and to the BM corpus guidelines. The simplifications of the BM corpus annotation scheme, compared to the scheme used in the training corpora, motivated our approach to relaxing the constraints used to link arguments to event triggers, as described in the previous section.

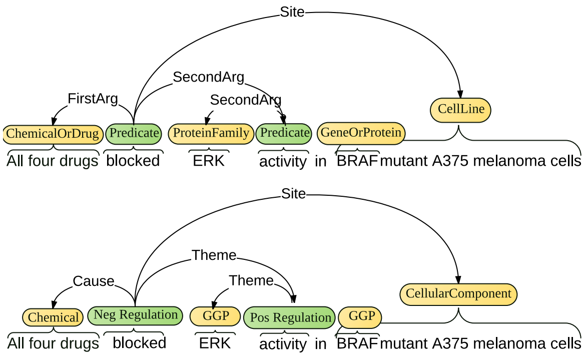

Figure 2: Example of event annotation in the BM corpus (top) versus BioNLP (bottom) : we can observe the different labels used for entities, events and edges

Due to the small size of BM corpus presented above, our experiments were repeated on the MLEE corpus (Pyysalo et al., 2012), using the development set as a test case. The MLEE corpus was chosen because as BM it displays a wide range of entities and events that spanned across different levels of biological organisation (molecular to organ) instead of focusing on protein reactions. The experiments were repeated twice, once including the MLEE training corpus in the training data, and once keeping it only for test purposes, in order to provide a comparison between application on seen and unseen data (see Section 6). For the purposes of this study, the edge annotations in MLEE are simplified to *Arg1* and *Arg2*, such that it is in line with the annotation scheme of the BM corpus, allowing for the relaxed constraint approach to be applied and for a better comparison of the results.

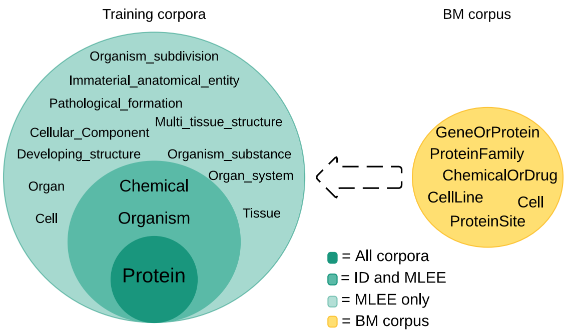

Figure 3: Initial named entity labels for the corpora used in the experiments. BM corpus labels were adapted to the training corpora as explained in section 4.1

4 Methodology

4.1 Adapting the entity labels without supervision

Since there is no widely accepted standard in the community in terms of the annotation labels for named entities, a common issue when processing multiple corpora, is overlapping annotations. In other words, different labels may be used to describe the same entity type, and that exactly is the case for the BM corpus when compared to the training corpora.[3] Hence, when testing on unseen data, it is necessary to map the labels of the test corpus to the ones that the model is trained to recognise, in order to obtain optimal results. For example, proteins were annotated as *GeneOrProtein* in the BM corpus and as *Protein* in the training corpora. For the filtering and unification of annotations instead of manually identifying the overlapping annotations, a heuristic automated label filtering method was implemented, in order to map the labels of the target/test corpus (TL) to those of the source/training one (SL). To that end, label similarity was calculated based on the following heuristic formula:

[3] The training corpora also contained conflicting / overlapping annotations initially that were priorly resolved in a similar manner

$$TL_i \rightarrow SL_j,$$

$$SL_j \rightarrow \underset{k}{\mathrm{argmax}}\left(\frac{\#(AnnE\_TL_i \cap AnnE\_SL_k)}{\#AnnE\_SL_k}\right) \tag{1}$$

where $AnnE\_TL_i$ corresponds to an annotated text span under the label TL_i in the target corpus, while $AnnE\_SL_j$ to an annotated text span under the label SL_j. The aforementioned text spans, can be single or multi-word tokens. Using this method, each label from the test corpus was assigned to the most similar label in the training corpus. For BM the labels were adapted as following:

BM corpus		Training Corpora
GeneOrProtein	→	*Protein*
ProteinFamily	→	*Protein*
ProteinSite	→	*Protein*[4]
ChemicalOrDrug	→	*Chemical*
CellLine	→	*Cellular_component*
SubcellularLocation	→	*Protein*

Table 1: Mapping between BM and trining corpora named entity annotations

It should be mentioned that in the case of the BM corpus, while there were obviously synonymous labels, the actual overlap found by the above technique was less than 10% for all labels, nonetheless still valid. In general this technique allows corpora to be added or removed without the need to fully revise the corpus. The same method could be used for different annotation types, such as event or edge type annotation.

4.2 Re-evaluating argument candidates of correct partial events

We hypothesise that, owing to the complexity of event patterns sought by the model, it sometimes fails to identify the complete set of arguments for an event, even if those arguments are correctly identified in the text as entities. This leads to the identification of partial events, such as the one presented in Figure 4. In order to identify the missing arguments, we aim to reduce the complexity of learned patterns, while complying with the annotation of the BM corpus.

Thus, we apply a relaxed post-processing step to the event extraction results, such that constraints regarding the learned roles of arguments are no longer imposed. Approaches that relax rule or pattern constraints have previously been shown to constitute an efficient method of achieving generalisation and allowing models to be better adapted to other natural language processing tasks such as named entity recognition in (Tatar and Cicekli, 2011) and (Zhou and Su, 2003) or information extraction in (Ciravegnia and Lavelli, 2004). In our case, the relaxed constraints permit a re-evaluation of the possible relations between event triggers and recognised entities in a given sentence.

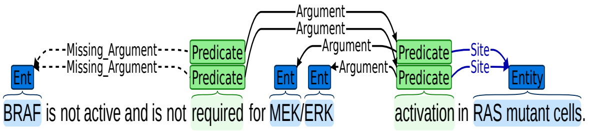

Figure 4: Example of partial event from the test corpus: *BRAF* should also be linked to the event but is missed by the EventMine model

We implemented a ranking approach such that for each identified event trigger, all the entities in the same sentence that are not already linked to it by the EventMine model are ranked according to their likelihood of being related to the trigger.

The entity ranking is based on syntactic dependencies between word tokens for each sentence. The underlying assumption is that for an entity to be linked to an event trigger as an argument, there has to be some syntactic relation between the two terms. Syntactic analysis is undertaken by the Enju syntactic parser (Miyao et al., 2008), which has a model trained on biomedical corpora. Since each dependency can be seen as a link between two words, we can consider dependency relations as structured dependency graphs. Dependency graphs have been used before in event extraction (Buyko et al., 2009; Liu et al., 2013) with Liu's approach, on subgraph matching of directed dependency graphs, achieving high precision but low recall. Aiming for high recall, we take a different approach; in the undirected graphs, we expect the path between a trigger word and its related arguments to be shorter than the path between the same trigger word and other, non related entities in the same sentence. We thus consider the shortest dependency path length as the main feature for ranking. For example from the Enju output for the partial event shown in Figure 5, we can see that the shortest dependency path length between the entity BRAF and the event trigger is equal to 1 (direct link). To facilitate full exploitation of the dependency graph, we considered the following:

[4] No overlap found, manual decision.

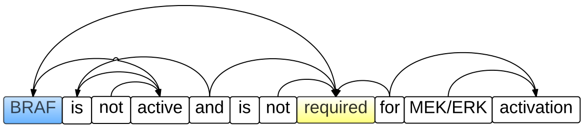

Figure 5: Dependency link representation example for a Biomedical sentence as analysed by Enju

- **Dependency type:** Enju provides the dependency type (prepositional, coordination, noun modifier, etc) for each dependency link/edge. This type can be exploited either to assign different weights to each dependency edge of the argument-trigger path, or to consider different path patterns. Such manipulation has been employed in rule-based event extraction in (Kilicoglu and Bergler, 2009), achieving good accuracy but low recall. Also, extracting specific path patterns renders the approach dependent on a particular parser, thus limiting the independence and adaptability of its application. Since the focus of this study was on recall and adaptability, the dependency type information was ignored, except for the case that follows.
- **Flattening coordination:** We decided that preprocessing was necessary to resolve coordination dependencies, such that a given entity would have the same distance to a trigger, regardless of the existence of a coordination argument dependency. Accordingly, in the calculation of the shortest paths, all coordination dependencies (labelled as *coord_arg* by Enju) are flattened as shown in Figure 6.

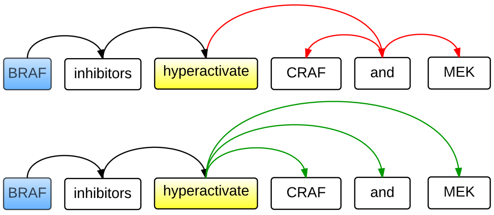

Figure 6: Flattening coordination dependencies

- **Nested Events:** To handle nested events, extracted events were also considered as entities, using the trigger as the representative text span, used to calculate the distance to the trigger of the top-level event and the rest of the features.

In order to obtain the rankings, we firstly consider our problem as one of *binary classification*, where the task is to classify each entity with respect to each trigger, as a valid (positive case) or non-valid (negative case) argument. Then, in training a classifier on the above binary classification task, we can employ the prediction confidence of the classifier model in order to rank the entities with respect to the event. The top ranked entity is selected and added to the event. In order to train a strong classifier model, a greater number of attributes that indicate the relation of an entity to a trigger were considered and implemented as additional features for the classifier. The main feature classes of the final feature set are listed below:

- Shortest dependency path (numeric)
- Entity Type (nominal)
- Participation in other events (binary)
- PoS (Part of Speech) (nominal)
- Context PoS (surrounding tokens) (nominal)
- Relative position to the event trigger (before/after) (nominal)
- Dependency on a prepositional token - type of prepositional token (binary-nominal)
- Event type (nominal)
- Token distance to trigger (numeric).

For the binary classification task, after comparison of an SVM, a logistic regression and an Adaboost classifier (implemented with random tree models), the AdaBoost classifier was chosen as it outperformed the rest by at least 10% (10 fold cross validation F-score on training set: 0.93). [5]

We also tried to avoid the addition of spurious arguments to events. Our initial experiments revealed that a considerable number of events require either a single argument or no arguments. For some event types such as *Gene_Expression*, such cases constituted more than 80% of the events. In order to avoid the addition of spurious arguments, and inspired by (Rahman and Ng, 2009) , an artificial *"null"* named entity instance was created for each event, and assigned to the events in the training set that did not require a second (or even a first) argument. Thus, the classifier would consider and rank the null entity along with the rest for each event.

Finally, to account for entities that are indirectly linked to events, i.e. those which occur as arguments of nested events, for each trigger, entities

[5] It should be noted that, while Adaboost appears to be most efficient for the purposes of our study, our classification task is only binary, and it is not straightforward to assume that it would outperform SVM in the rest of the EventMine pipeline, without additional testing

belonging to its parent event or its nested events were excluded from ranking. Furthermore, entities that were already assigned to a different event having the same trigger, as in Figure 7, were considered mutually exclusive.

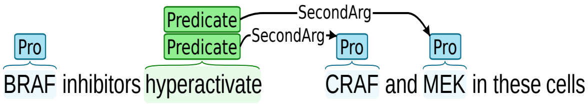

Figure 7: CRAF and MEK are mutually exclusive

5 Evaluation

In order to evaluate the performance of our method we compare the identified arguments with the ones annotated in the gold corpus.

For each event annotated by EventMine, we define argument recall and precision as :

$$Recall = \frac{arg_{EM} \cap arg_{gold}}{arg_{gold}} \quad (2)$$

$$Precision = \frac{arg_{EM} \cap arg_{gold}}{arg_{EM}} \quad (3)$$

where arg_{EM} is the set of arguments EventMine identifies for this event and arg_{gold} the corresponding set of arguments identified in the gold standard. [6]

6 Results and Discussion

6.1 Experimental Results

We applied and evaluated the ranking methodology to the corpora described in Section 3 and the results are shown in Tables 2 and 3.

	Precision		Recall		Fscore		Percent.
	EM	+R	EM	+R	EM	+R	in corpus
Phosphorylation	0.93	0.82	0.86	0.93	0.89	0.87	10
Planned_process	0.5	0.5	0.5	0.5	0.50	0.50	2
Negative_regulation	0.91	0.86	0.83	0.83	0.87	0.84	16
Localization	1	1	0.67	0.67	0.80	0.80	1
Regulation	0.88	0.88	0.88	0.88	0.88	0.88	3
Gene_expression	0.8	0.8	0.75	0.75	0.77	0.77	7
Binding	0.47	0.47	0.43	0.47	0.45	0.47	2
Positive_regulation	0.66	0.59	0.61	0.63	0.63	0.61	44
Total	0.67	0.63	0.62	0.64	0.64	0.63	

Table 2: Results on BM corpus before (EM) and after Re-ranking (+R)

For both corpora, our ranking method leads to an increase in recall compared to the default EventMine application. However, our method also results in a decrease in precision. In order to appreciate the impact of missing arguments on unseen data we repeat the experiment using the MLEE training corpus during training for EventMine. In this case (see Table 4), the recall is higher and the improvement from the post-processing step not significant, suggesting that the post-processing methodology is advantageous mostly when EventMine is applied to new domains.

	Precision		Recall		Fscore		Percent.
	EM	+R	EM	+R	EM	+R	in corpus
Protein_catabolism	0.5	0.5	0.5	0.5	0.50	0.50	2
Phosphorylation	0.69	0.59	0.69	0.69	0.69	0.64	5
Dissociation	0.78	0.44	1	1	0.88	0.61	1
Transcription	0.5	0.5	0.5	0.5	0.50	0.50	2
Negative_regulation	0.5	0.48	0.38	0.5	0.43	0.49	9
Regulation	0.53	0.43	0.4	0.47	0.46	0.45	4
Gene_expression	0.88	0.88	0.86	0.86	0.87	0.87	27
Localization	0.63	0.61	0.69	0.76	0.66	0.68	6
Positive_regulation	0.72	0.65	0.63	0.67	0.67	0.66	32
Binding	0.66	0.68	0.56	0.64	0.61	0.66	11
Total	0.7	0.67	0.64	0.68	0.67	0.67	

Table 3: Results on MLEE corpus before (EM) and after Re-ranking (+R)

	Precision		Recall		Fscore		Percent.
	EM	+R	EM	+R	EM	+R	in corpus
Protein_catabolism	0.6	0.6	0.6	0.6	0.60	0.60	1
Death	0.71	0.71	0.71	0.71	0.71	0.71	1
Transcription	0.5	0.5	0.5	0.5	0.50	0.50	1
Localization	0.76	0.67	0.77	0.77	0.76	0.72	5
Development	0.55	0.55	0.55	0.55	0.55	0.55	4
Regulation	0.49	0.45	0.45	0.46	0.47	0.45	7
Breakdown	0.78	0.78	0.78	0.78	0.78	0.78	1
Positive_regulation	0.68	0.64	0.63	0.64	0.65	0.64	22
Growth	0.88	0.88	0.88	0.88	0.88	0.88	3
Phosphorylation	0.69	0.56	0.69	0.69	0.69	0.62	2
Blood_vessel_development	0.96	0.96	0.75	0.75	0.84	0.84	16
Dissociation	0.67	0.42	1	1	0.80	0.59	1
Cell_proliferation	0.92	0.92	0.92	0.92	0.92	0.92	1
Pathway	0.54	0.46	0.46	0.48	0.50	0.47	1
Planned_process	0.81	0.77	0.7	0.71	0.75	0.74	10
Negative_regulation	0.76	0.7	0.66	0.67	0.71	0.68	10
Gene_expression	0.88	0.88	0.88	0.88	0.88	0.88	9
Binding	0.73	0.71	0.65	0.68	0.69	0.69	4
Tissue_remodeling	1	1	1	1	1.00	1.00	1
Total	0.76	0.73	0.68	0.69	0.72	0.71	

Table 4: Results on MLEE corpus before (EM) and after Re-Ranking (+R) (MLEE training set added to the training corpora of EventMine)

Moreover, we can observe in Tables 3 and 4 that the event types recognised are not 100% overlapping. Indeed, since in the first case EventMine is not trained on the MLEE corpus, the set of event types that it is trained to recognise only partially overlaps with the event types annotated in MLEE. As such, in a large number of cases, even though the event trigger is correctly extracted, it is attributed an event type other than the one annotated in the gold standard. For example some of the *BreakDown* events (in Table 4) tend to be recognised as *Negative Regulation* when the model is not trained on the MLEE events (Table 3). We thus wanted to examine the impact of erroneous

[6] For events that are not matched in the gold standard both values are zero.

event type identification on the linking and precision of argument linking, given that EventMine models learn different annotation constraints for each event type. Table 5 compares the performance achieved by our method when the event type assigned by EventMine matches the label in the gold standard, with the overall performance. It can be observed that when the labels do match the performance increases significantly. Thus, it seems that part of error in linking arguments to an event derives from an erroneous recognition of the type of the argument, that is often linked to events that the model is not trained to recognise properly.

	Overall Results		Same label in GS		
	Precision	Recall	Precision	Recall	Percentage
Protein_catabolism	0.5	0.5	1	1	0.67
Phosphorylation	0.69	0.69	1	1	1
Dissociation	0.78	1	1	1	0.33
Transcription	0.5	0.5	1	1	1
Negative_regulation	0.5	0.38	1	0.75	0.93
Localization	0.53	0.4	0.89	0.67	0.93
Gene_expression	0.88	0.86	1	1	0.91
Regulation	0.63	0.69	0.79	0.87	1
Binding	0.72	0.63	0.93	0.81	1
Positive_regulation	0.66	0.56	0.87	0.75	0.94

Table 5: Performance of EventMine for matching type annotations versus overall results

6.2 Analysis of Results and Performance Considerations

The results shown in the previous section are promising in terms of recall. However, there is still considerable room for improvement, especially in terms of decreasing the added noise, so as to minimise the drop in precision. Below we present the most important observations regarding our results and we analyse the errors produced.

- **Correct identification of the partial event but erroneous identification of the missing argument:** Of the noisy events, 60% constituted cases that were correctly identified as partial events, but where the ranking algorithm failed to identify the correct entity to link to the trigger. This was a common pattern in cases where the argument was an event, but the ranking system actually selected one of that event's arguments instead of the whole event, as illustrated in Figure 8. It is important to note that in some of such cases, the event trigger was not annotated by EventMine in the first place. Thus, it was impossible for our method to capture it. This emphasises the strong dependency of our method on EventMine's performance. A possible solution to this problem, which will be considered as future work, is to reformulate the problem as a joint learning task, in which one classifier would focus on ranking single named entity candidates and the other on ranking event candidates, and they would be combined in the test corpus in order to choose the most likely solution. Such an approach would, however, have increased complexity, and its results remain to be tested. [7]

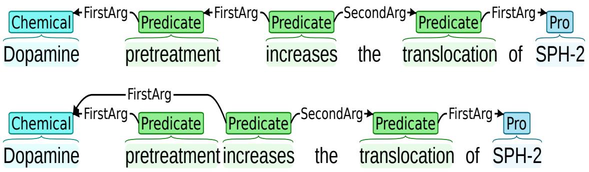

Figure 8: Linking the nested event argument instead of the trigger: Compare correct annotation (top) with produced one (bottom)

- **Entities related to the event in a complementary manner:** In a considerable number of erroneous cases, the ranking system identified arguments that were not annotated in the gold corpus, but which nevertheless were related to the trigger. Two distinctive patterns emerged, as illustrated in Figure 9.

  1. Aliases of the original argument, used in the same sentence (usually a superclass)
  2. Text spans with multiple annotations that are linked multiple times to the event as separate entities

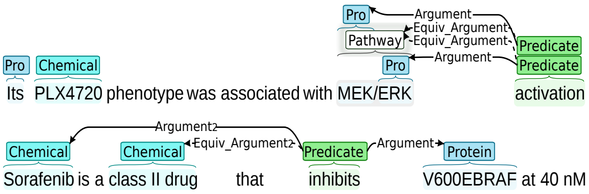

Figure 9: Multi annot. (top): Pathway entity erroneously considered a valid additional argument Alias (bottom): Sorafenib and its superclass both considered valid argument candidates that are not mutually exclusive

- **Overfitting to "null" instances:** As can be deduced from the result tables (2, 3 and 4), there was a considerable percentage of partial events whose missing arguments were still not fully identified by our method. In those cases, the classifier ranked the "null" instance mentioned

[7] However the performance will still depend on the recall of the event extraction system.

in Section 4 as the best option. Investigation revealed that for partial events, the correct missing argument was ranked second after the “null” instance in more than 50% of cases (null instance suggestions accounted for 80%-70% of the total suggestions). A possible solution to further increase the recall would be to drop the “null” instance implementation, and use a confidence threshold instead. However, such a method would be more *ad hoc*, having severe implications on the generalisability of the model.

As a final note, it should be mentioned that in some cases, our method made suggestions that could correct events containing errors (i.e., correct trigger but wrong argument). While these cases were not considered in the scope of this work, it would be interesting to investigate how our method could be adapted/expanded to suggest argument corrections as well as additions.

7 Conclusions and Future Work

Our novel approach to improving event extraction results has successfully shown that identifying and ranking additional arguments by relaxing annotation constraints can aid in improving the argument recall and reducing partial (and sometimes even erroneous) event extraction. Of particular note is the demonstration that our approach has the greatest impact when applied to unseen data. As such, we consider that our results are extremely promising, even though there is still a large margin for further improvements and experimentation.

An important feature of our approach is that the methodology employed is generic enough to be applied to output of any other event extraction architecture (particularly pipelined ones) or any other biomedical corpus without significant modification. Future testing on different corpora and annotation schemes will help to reinforce the robustness and generalisability of our method.

However, this study has already revolved various promising areas for further investigation, in terms of both increasing recall and reducing noisy additions. Of particular interest would be to see whether employing methods with multiple classifiers (co-training, joint-learning or ensemble methods) would improve the performance and reduce the noise. Such an approach could target either classifiers trained on different argument types (named entities or entire events) or even classifiers specialising in particular event types. However, this would constitute a whole new area of research and experimentation.

A further aspect, only minimally considered in this work, is the influence of the training instances and labels on the performance. On the MLEE corpus, it was observed that for events whose automatically assigned event type did not match the gold standard, argument recall and precision also deteriorated. Hence, we can deduce that improving the accuracy of event type assignment would have a positive impact on event extraction performance. The same conclusion could hold also for the named entity labels; as mentioned in Section 4, the BM corpus was initially annotated with a different NE label-set that was automatically (without supervision) aligned with the training corpus annotations in order for the trained model to be applied to it. However, instead of adapting the testing corpus annotations, it would be worthwhile to provide efficient unsupervised methods for adapting the labels in the training corpus to those in the testing corpus. Such an approach could boost the precision without compromising recall by reducing the impact of training on instances (events) that are not related to the ones in the test set. To that end, it would be interesting to combine the wide coverage approach (Miwa et al., 2013) with domain adaptation approaches such as the ones mentioned in Section 2.3 or simply instance reweighting ones such as (Jiang and Zhai, 2007).

The above considerations will be vital in facilitating the incorporation of constraint relaxation as an integral part of the EventMine architecture, rather that as a post-processing step. This will help to enhance EventMine’s properties of generalisability and adaptability, and thus allow it it to achieve more robust performance. However, the challenge will be to consider the constraint relaxation and adaptation problem globally, rather than only for argument role annotation constraints.

Acknowledgments

This work was supported by the DARPA funded *Big Mechanism* Project, as well as by the EPSRC funded *Centre for Doctoral Training in Computer Science* scholarship. We would like to thank Dr. Riza Theresa Batista-Navarro and Dr. Ioannis Korkontzelos for the useful discussions and feedback at critical points. Finally, we would like to thank our referees for their constructive input.

References

Sophia Ananiadou, Paul Thompson, Raheel Nawaz, John McNaught, and Douglas B Kell. 2014. Event-based text mining for biology and functional genomics. *Briefings in functional genomics*, page elu015.

Jari Björne and Tapio Salakoski. 2011. Generalizing biomedical event extraction. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, BioNLP Shared Task '11, pages 183–191, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jari Björne and Tapio Salakoski. 2013. Tees 2.1: Automated annotation scheme learning in the bionlp 2013 shared task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 16–25.

Jari Björne, Filip Ginter, Sampo Pyysalo, Jun'ichi Tsujii, and Tapio Salakoski. 2010. Complex event extraction at pubmed scale. *Bioinformatics*, 26(12):i382–i390.

Ekaterina Buyko, Erik Faessler, Joachim Wermter, and Udo Hahn. 2009. Event extraction from trimmed dependency graphs. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 19–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

Etienne Caron, Samik Ghosh, Yukiko Matsuoka, Dariel Ashton-Beaucage, Marc Therrien, Sébastien Lemieux, Claude Perreault, Philippe P Roux, and Hiroaki Kitano. 2010. A comprehensive map of the mtor signaling network. *Molecular systems biology*, 6(1).

Minmin Chen, Kilian Q Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. In *Advances in neural information processing systems*, pages 2456–2464.

F. Ciravegnia and A. Lavelli. 2004. Learningpinocchio: adaptive information extraction for real world applications. *Natural Language Engineering*, 10:145–165, 6.

Paul R. Cohen. 2014. Darpa's big mechanism program. `http://www.darpa.mil/Our_Work/I2O/Programs/Big_Mechanism.aspx`.

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

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

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *ACL*, volume 7, pages 264–271.

Halil Kilicoglu and Sabine Bergler. 2009. Syntactic dependency based heuristics for biological event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 119–127. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

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 the BioNLP Shared Task 2011 Workshop*, pages 1–6. Association for Computational Linguistics.

B. Kulis, K. Saenko, and T. Darrell. 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792, June.

Haibin Liu, Lawrence Hunter, Vlado Kešelj, and Karin Verspoor. 2013. Approximate subgraph matching-based literature mining for biomedical events and relations. *PloS one*, 8(4):e60954.

Makoto Miwa and Sophia Ananiadou. 2013. Nactem eventmine for bionlp 2013 cg and pc tasks. In *Proceedings of BioNLP Shared Task 2013 Workshop*, pages 94–98.

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

Yusuke Miyao, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *ACL*, volume 8, pages 46–54.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7.

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

Sinno Jialin Pan, I.W. Tsang, J.T. Kwok, and Qiang Yang. 2011. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on*, 22(2):199–210, Feb.

Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, and Junichi Tsujii. 2011. Towards exhaustive event extraction for protein modifications. In *Proceedings of BioNLP 2011 Workshop*, pages 114–123.

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

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 968–977. Association for Computational Linguistics.

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

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Serhan Tatar and Ilyas Cicekli. 2011. Automatic rule learning exploiting morphological features for named entity recognition in turkish. *Journal of Information Science*, 37(2):137–151.

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

Rui Xia, Chengqing Zong, Xuelei Hu, and E. Cambria. 2013. Feature ensemble plus sample selection: Domain adaptation for sentiment classification. *Intelligent Systems, IEEE*, 28(3):10–18, May.

Guodong Zhou and Jian Su. 2003. System for recognising and classifying named entities, December 31. US Patent App. 10/585,235.