Biomedical Event Extraction without Training Data

Andreas Vlachos, Paula Buttery, Diarmuid Ó Séaghdha, Ted Briscoe
Computer Laboratory
University of Cambridge
Cambridge, UK
av308,pjb48,do242,ejb@cl.cam.ac.uk

Abstract

We describe our system for the BioNLP 2009 event detection task. It is designed to be as domain-independent and unsupervised as possible. Nevertheless, the precisions achieved for single theme event classes range from 75% to 92%, while maintaining reasonable recall. The overall F-scores achieved were 36.44% and 30.80% on the development and the test sets respectively.

1 Introduction

In this paper we describe the system built for the BioNLP 2009 event detection and characterization task (Task 1). The approach is based on the output of a syntactic parser and standard linguistic processing, augmented by rules acquired from the development data. The key idea is that a trigger connected with an appropriate argument along a path through the syntactic dependency graph forms an event.

The goal we set for our approach was to avoid using training data explicitly annotated for the task and to preserve domain independence. While we acknowledge the utility of supervision (in the form of annotated data) and domain knowledge, we believe it is valuable to explore an unsupervised approach. Firstly, manually annotated data is expensive to create and the annotation process itself is difficult and unavoidably results in inconsistencies, even in well-explored tasks such as named entity recognition (NER). Secondly, unsupervised approaches, even if they fail to reach the performance of supervised ones, are likely to be informative in identifying useful features for the latter. Thirdly, exploring the potential of such a system may highlight what domain knowledge is useful and its potential contribution to performance. Finally, preserving domain independence allows us to develop and evaluate a system that could be used for similar tasks with minimal adaptation.

The overall architecture of the system is as follows. Initiallly, event triggers are identified and labelled with event types using seed terms. Based on the dependency output of the parser the triggers are connected with candidate arguments using patterns identified in the development data. Anaphoric candidate arguments are then resolved. Finally, the triggers connected with appropriate arguments are post-processed to generate the final set of events. Each of these stages are described in detail in subsequent sections, followed by experiments and discussion.

2 Trigger identification

We perform trigger identification using the assumption that events are triggered in text either by verbal or nominal prdicates (Cohen et al., 2008).

To build a dictionary of verbs and their associated event classes we use the triggers annotated in the training data. We lemmatize and stem the triggers with the morphology component of the RASP toolkit (Briscoe et al., 2006)[1] and the Porter stemmer[2] respectively. We sort the trigger stem - event class pairs found according to their frequency in the training data and we keep only those pairs that appear at least 10 times. The trigger stems are then mapped to verbs. This excludes some relatively common triggers, which will reduce recall, but, given that we rely exclusively on the parser for

[1] http://www.cogs.susx.ac.uk/lab/nlp/rasp/

[2] http://www.tartarus.org/˜martin/PorterStemmer

argument extraction, such triggers would be difficult to handle. For verbs with more than one event class we keep only the most frequent one.

We consider the assumption that each verb denotes a single event class to be a reasonable one given the restricted task domain. It hinders us from dealing with triggers denoting multiple event classes but it simplifies the task so that we do not need annotated data. While we use the training data triggers to obtain the list of verbs and their corresponding event types, we believe that such lists could be obtained by clustering (Korhonen et al., 2008) with editing and labelling by domain experts. This is the only use of the training data we make in our system.

During testing, using the tokenized text provided, we attempt to match each token with one of the verbs associated with an event type. We perform this by relaxing the matching successively, using the token lemma, then stem, and finally allowing a partial match in order to deal with particles (so that e.g. *co-transfect* matches *transfect*). This process returns single-token candidate triggers which, while they do not reproduce the trigger annotation, are likely to be adequate for event extraction. We overgenerate triggers, since not all occurrences denote an event, either because they are not connected with appropriate arguments or because they are found in a non-event denoting context, but we expect to filter these at the argument extraction stage.

3 Argument extraction

Given a set of candidate triggers, we attempt to connect them with appropriate arguments using the dependency graph provided by a parser. In our experiments we use the domain-independent unlexicalized RASP parser, which generates parses over the part-of-speech (PoS) tags of the tokens generated by an HMM-based tagger trained on balanced English text. While we expect that a parser adapted to the biomedical domain may perform better, we want to preserve the domain-independence of the system and explore its potential.

The only adjustment we make is to change the PoS tags of tokens that are part of a protein name to proper names tags. We consider such an adjustment domain-independent given that NER is available in many domains (Lewin, 2007). Following Haghighi et al (2005), in order to ameliorate parsing errors, we use the top-10 parses and return a set of bilexical head-dependent grammatical relations (GRs) weighted according to the proportion and probability of the top parses supporting that GR.

The GRs produced by the parser define directed graphs between tokens in the sentence, and a partial event is formed when a path that connects a trigger with an appropriate argument is identified. GR paths that are likely to generate events are selected using the development data, which does not contradict the goals of our approach because we do not require annotated training data. Development data is always needed in order to build and test a system, and such supervision could be provided by a human expert, albeit not as easily as for the list of trigger verbs. The set of GR paths identified follow:

VERB-TRIGGER –subject– ARG
NOUN-TRIGGER –iobj– PREP –dobj– ARG
NOUN-TRIGGER –modifier– ARG
TRIGGER –modifier– PREP –obj– ARG
TRIGGER –passive subject– ARG

The final system uses three sets of GR paths: one for Regulation events; one for Binding events; and one for all other events. The difference between these sets is in the lexicalization of the linking prepositions. For example, in Binding events the linking preposition required lexicalization since *binds x to/with y* denotes a correct event but not *binds x by y*. Binding events also required additional GR paths to capture constructions such as *binding of x to y*. For Regulation events, the path set was further augmented to differentiate between theme and cause. When the lexicalized GR pattern sets yielded no events we backed-off to the unlexicalized pattern set, which is identical for all event types. In all GR path sets, the trigger was unlexicalized and only restricted by PoS tag.

4 Anaphora resolution

The events and arguments identified in the parsed abstracts are post-processed in context to identify protein referents for event arguments that are anaphoric (e.g., *these proteins*, *its phosphorylation*) or too complex to be extracted directly from the grammatical relations (*phosphorylation of cellular proteins , notably phospholipase C gamma 1*). The

anaphoric linking is performed by a set of heuristic rules manually designed to capture a number of common cases observed in the development dataset. A further phenomenon dealt with by rules is coreference between events, for example in *The expression of LAL-mRNA is induced. This induction is dependent on...* where the Induction event described by the first sentence is the same as the theme of the Regulation event in the second and should be given the same event index. The development of the post-processing rules favoured precision over recall, but the low frequency of each case considered means that some overfitting to the development data may have been unavoidable.

5 Event post-processing

At the event post-processing stage, we form complete events considering the trigger-argument pairs produced at the argument extraction stage whose arguments are resolved (possibly using anaphora resolution) either to a protein name or to a candidate trigger. The latter are considered only for regulation event triggers. Furthermore, regulation event trigger-argument pairs are tagged either as theme or cause at the argument extraction stage.

For each non-regulation trigger-argument pair, we generate a single event with the argument marked as theme. Given that we are dealing only with Task 1, this approach is expected to deal adequately with all event types except Binding, which can have multiple themes. Regulation events are formed in the following way. Given that the cause argument is optional, we generate regulation events for trigger-argument pairs whose argument is a protein name or a trigger that has a formed event. Since regulation events can have other regulation events as themes, we repeat this process until no more events can be formed. Occasionally, the use of multiple parses results in cycles between regulation triggers which are resolved using the weighted GR scores. Then, we attach any cause arguments that share the same trigger with a formed regulation event.

In the analysis performed for trigger identification in Section 2, we observed that certain verbs were consistently annotated with two events (namely *overexpress* and *transfect*), a non-regulation event and a regulation event with the former event as its theme. For candidate triggers that were recognized due to such verbs, we treat them as non-regulation events until the post-processing stage where we generate two events.

6 Experiments - Discussion

We expected that our approach would achieve high precision but relatively low recall. The evaluation of our final submissions on the development and test data (Table 1) confirmed this to a large extent. For the non-regulation event classes excluding Binding, the precisions achieved range from 75% to 92% in both development and test data, with the exception of Transcription in the test data. Our approach extracts Binding events with a single theme, more suitably evaluated by the Event Decomposition evaluation mode in which a similar high precision/low recall trend is observed, albeit with lower scores.

Of particular interest are the event classes for which a single trigger verb was identified, namely Transcription, Protein catabolism and Phosphorylation, which makes it easier to identify the strengths and weaknesses of our approach. For the Phosphorylation class, almost all the triggers that were annotated in the training data can be captured using the verb *phosporylate* and as a result, the performances achieved by our system are 70.59% and 60.63% F-score on the development and test data respectively. The precision was approximately 78% in both datasets, while recall was lower due to parser errors and unresolved anaphoric references. For the Protein catabolism class, *degrade* was identified as the only trigger verb, resulting in similar high precision but relatively lower recall due to the higher lexical variation of the triggers for this class. For the Transcription class we considered only *transcribe* as a trigger verb, but while the performance on the development data is reasonable (55%), the performance on the test data is substantially lower (20%). Inspecting the event triggers in the training data reveals that some very common triggers for this class either cannot be mapped to a verb (e.g., *mrna*) or are commonly used as triggers for other event classes. A notable case of the latter type is the verb *express*, which, while mostly a Gene Expressions trigger, is also annotated as Transcription more than 100 times in the training data. Assuming that this is desirable,

	Development			Test		
Event Class	recall	precision	fscore	recall	precision	fscore
Localization	45.28	92.31	60.76	25.86	90.00	40.18
Binding	12.50	24.41	16.53	12.68	31.88	18.14
Gene expression	52.25	80.79	63.46	45.57	75.81	56.92
Transcription	42.68	77.78	55.12	12.41	56.67	20.36
Protein catabolism	42.86	81.82	56.25	35.71	83.33	50.00
Phosphorylation	63.83	78.95	70.59	49.63	77.91	60.63
Event Total	39.03	65.97	49.05	33.16	68.15	44.61
Regulation	20.12	50.75	28.81	9.28	36.49	14.79
Positive regulation	16.86	48.83	25.06	11.39	38.49	17.58
Negative regulation	11.22	36.67	17.19	6.86	36.11	11.53
Regulation Total	16.29	47.06	24.21	9.98	37.76	15.79
Total	26.55	58.09	36.44	21.12	56.90	30.80
Binding (decomposed)	26.92	66.14	38.27	18.84	54.35	27.99

Table 1: Performance analysis on development and test data using Approximate Span/Partial Recursive Matching.

a more appropriate solution would need to take context into account.

Our performance on the regulation events is substantially lower in both recall and precision. This is expected, as they rely on the extraction of non-regulation events. The variety of lexical triggers is not causing the drop in performance though, since our system performed reasonably well in the Gene Expression and Localization classes which have similar lexical variation. Rather it is due to the combination of the lexical variation with the requirement to make the distinction between the theme and optional cause argument, which cannot be handled appropriately by the small set of GR paths employed.

The contribution of anaphora resolution to our system is limited as it relies on the argument extraction stage which, apart from introducing noise, is geared towards maintaining high precision. Overall, it contributes 22 additional events on the development set, of which 14 out of 16 are correct non-regulation events. Of the remaining 6 regulation events only 2 were correct. Similar trends were observed on the test data.

7 Conclusions - Future work

We described an almost unsupervised approach for the BioNLP09 shared task on biomedical event extraction which requires only a dictionary of verbs and a set of argument extraction rules. Ignoring trigger spans, the performance of the approach is parser-dependent and while we used a domain-independent parser in our experiments we also want to explore the benefits of using an adapted one.

The main weakness of our approach is the handling of events with multiple arguments and the distinctions between them, which are difficult to deal with using simple unlexicalized rules. In our future work we intend to explore semi-supervised approaches that allow us to acquire more complex rules efficiently.

References

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL Interactive presentation sessions*, pages 77–80.

Kevin B. Cohen, Martha Palmer, and Lawrence Hunter. 2008. Nominalization and alternations in biomedical language. *PLoS ONE*, 3(9).

Aria Haghighi, Kristina Toutanova, and Chris Manning. 2005. A Joint Model for Semantic Role Labeling. In *Proceedings of CoNLL-2005: Shared Task*.

Anna Korhonen, Yuval Krymolowski, and Nigel Collier. 2008. The choice of features for classification of verbs in biomedical texts. In *Proceedings of Coling*.

Ian Lewin. 2007. BaseNPs that contain gene names: domain specificity and genericity. In *Proceedings of the ACL workshop BioNLP: Biological, translational, and clinical language processing*, pages 163–170.