Extracting Biological Events from Text Using Simple Syntactic Patterns

Quoc-Chinh Bui, Peter M.A. Sloot
Computational Science, Informatics Institute
University of Amsterdam
Science Park 904, Amsterdam, The Netherlands
{c.buiquoc,p.m.a.sloot}@uva.nl

Abstract

This paper describes a novel approach presented to the BioNLP'11 Shared Task on GENIA event extraction. The approach consists of three steps. First, a dictionary is automatically constructed based on training datasets which is then used to detect candidate triggers and determine their event types. Second, we apply a set of heuristic algorithms which use syntactic patterns and candidate triggers detected in the first step to extract biological events. Finally, a post-processing is used to resolve regulatory events. We achieved an F-score of 43.94% using the online evaluation system.

1 Introduction

The explosive growth of biomedical scientific literature has attracted a significant interest on developing methods to automatically extract biological relations in texts. Until recently, most research was focused on extracting binary relations such as protein-protein interactions (PPIs), gene-disease, and drug-mutation relations. However, the extracted binary relations cannot fully represent the original biomedical data. Therefore, there is an increasing need to extract fine-grained and complex relations such as biological events (Miwa et al., 2010). The BioNLP'09 Shared Task (Kim et al., 2009) was the first shared task that provided a consistent data set and evaluation tools for extraction of such biological relations.

Several approaches to extract biological events have been proposed for this shared task. Based on their characteristics, these approaches can be divided into 3 groups. The first group uses a rule-based approach which implements a set of manually defined rules developed by experts or automatically learned from training data. These rules are then applied on dependency parse trees to extract biological events (Kaljurand et al., 2009; Kilicoglu and Bergler, 2009). The second group uses a machine learning (ML)-based approach which exploits various specific features and learning algorithms to extract events (Björne at al., 2009; Miwa et al., 2010). The third group uses hybrid methods that combine both rule- and ML-based approaches to solve the problem (Ahmed et al., 2009; Móra et al., 2009). Among these proposed approaches, the ML achieved the best results, however, it is non-trivial to apply.

In this paper, we propose a rule-based approach which uses two syntactic patterns derived from a parse tree. The proposed approach consists of the following components: a dictionary to detect triggers, text pre-processing, and event extraction.

2 System and method

2.1 Dictionary for event trigger detection

The construction of the dictionary consists of the following steps: grouping annotated triggers, filtering out irrelevant triggers, and calculating supportive scores. First, we collect all annotated triggers in the training and development datasets, convert them to lowercase format and group them based on their texture values and event types. For each trigger in a group, we count its frequency being annotated as trigger and its frequency being found in the training datasets to compute a confident score.

Next, we create a list of non-trigger words from the training dataset which consists of a list of prepositions (e.g. *to*, *by*), and a list of adjectives (e.g. *high*, *low*). We then filter out triggers that belong to the non-trigger list as well as triggers that consist of more than two words as suggested in the previous studies (Kilicoglu and Bergler, 2009). We further filter out more triggers by setting a frequency threshold for each event type. Triggers that

Proceedings of BioNLP Shared Task 2011 Workshop, pages 143–146,
Portland, Oregon, USA, 24 June, 2011.

have a frequency lower than a given threshold (which is empirically determined for each event type) are excluded.

In addition, for each binding trigger (i.e. trigger of binding event) we compute a *t2score* which is the ratio of having a second argument. For each regulatory trigger we compute an *escore* which is the ratio of having an event as the first argument (theme) and a *cscore* is the ratio of having a second argument (cause).

2.2 Text preprocessing

Text preprocessing includes splitting sentences, replacing protein names with place-holders, and parsing sentences using the Stanford Lexical Parser[1]. First, we split the input text (e.g. title, abstract, paragraph) into single sentences using LingPipe sentence splitter[2]. Sentences that do not contain protein names are dropped. Second, we replace protein names with their given annotated IDs in order to prevent the parser from segmenting multiple word protein names. Finally, the sentences are parsed with the Stanford parser to produce syntactic parse trees. All parse trees are stored in a local database for later use.

Detection of event trigger and event type: For each input sentence, we split the sentence into tokens and use the dictionary to detect a candidate trigger and determine its event type (hereafter we referred to as 'trigger' type). After this step, we obtain a list of candidate triggers and their related scores for each event type.

2.3 Event extraction

To extract the biological events from a parse tree after obtaining a list of candidate triggers, we adapt two syntactic patterns based on our previous work on extracting PPIs (Bui et al., 2011). These patterns are applied for triggers in noun, verb, and adjective form. In the following sections we describe the rules to extract events in more detail.

Rule 1*: Extracting events from a noun phrase (NP)*
If the candidate trigger is a noun, we find a NP which is a joined node of this trigger and at least one protein from the parse tree. There are two NP patterns that can satisfy the given condition which are shown in Figure 1. In the first case (form1), NP does not contain a PP tag, and in the second case (form2), the trigger is the head of this NP. Depending on the trigger type (simple, binding or regulatory event), candidate events are extracted by the following rules as shown in Table 1.

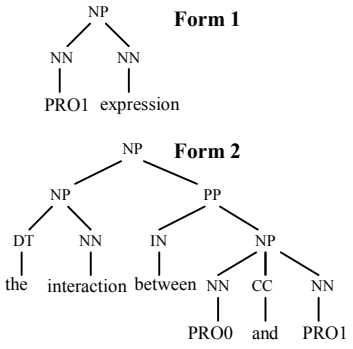

Figure 1: NP patterns containing trigger

Event type	Conditions and Actions
Simple or Regulatory	***NP in form1***: extract all proteins on the left of the trigger from NP. Form event pairs <trigger, protein>. ***NP in form2***: extract all proteins on the right of the trigger from NP. Form event pairs <trigger, protein>.
Binding	***NP in form1***: If proteins are in compound form i.e. PRO1/PRO2, PRO1-PRO2 then form an event triple <trigger, protein1, protein2>. Otherwise, form events pairs <trigger, protein>. ***NP in form2***: If NP contains one of the following preposition pairs: *between/and*, *of/with*, *of/to*, and the trigger' *t2score* >0.2 then split the proteins from NP into two lists: list1 and list2 based on the second PP (preposition phrase) or CC (conjunction). Form triples <trigger, protein1, protein2>, in which protein1 from list1 and protein2 from list2. Otherwise, form events the same way as simple event case.

Table 1: Conditions and actions to extract events from a NP. Simple and regulatory events use the same rules.

Rule 2*: Extracting events from a verb phrase (VP)*
If the candidate trigger is a verb, we find a VP which is a direct parent of this trigger from the parse tree and find a sister NP immediately preceding this VP. Next, candidate events are extracted by the following rules as shown in Table 2.

[1] http://nlp.stanford.edu/software/lex-parser.shtml
[2] http://alias-i.com/lingpipe/

The event trigger is an adjective: For a candidate trigger which is an adjective, if the trigger is in a compound form (e.g. PRO1-mediated), we apply *rule1* to extract events. In this case, the compound protein (e.g. PRO1) is used as *cause* argument. Otherwise, we apply *rule 2* to extract.

2.4 Post-processing

Post-processing includes determination of an event type for a shared trigger and checking cross-references of regulatory events. For each extracted event which has a shared trigger[3], this event is verified using a list of modified words (e.g. *gene*, *mRNA*) to determine final event type. For regulatory events, the post-processing is used to find cross reference events. The post-processing is shown in Algorithm 1.

Event type	Conditions and Actions
Simple	If VP contains at least one protein then extract all proteins which have a position on the right of the trigger from the VP to create a protein list. Otherwise, extract all proteins that belong to the NP. Form event pairs <trigger, protein> with the obtained protein list.
Binding	If VP contains at least one protein then extract all proteins which have a position on the right of the trigger from VP to create a protein list1. Extracting all proteins that belong to the NP to create protein list2. If both list1 and list2 are not empty then form triples <trigger, protein1, protein2>, in which protein1 from list1 and protein2 from list2. Otherwise, form event pairs <trigger, protein> from the non-empty protein list.
Regulatory	If trigger' *cscore*>0.3 then extract the same way as for the binding event, in which protein from list1 is used for cause argument. Otherwise follows the rule of the simple event.

Table 2: Conditions and actions to extract events from a VP

2.5 Algorithm to extract events

The whole process of extracting biological event is shown in Algorithm 1

Algorithm 1. // *Algorithm to extract biological events from sentence.*
Input: pre-processing sentence, parse tree, and lists of candidate triggers for each event type
Output: lists of candidate events of corresponding event type
Init: *found_list* = *null* // store extracted events for reference later

Step 1: Extracting events
For each event type
For each trigger of the current event type
Extract candidate events using extraction rules
If candidate event found
Store this event to the *found_list*
End if
End for
End for

Step 2: Post-preprocessing
For each extracted event from *found_list*
If event has a shared trigger
Verify this event with the modified words
If not satisfy
Remove this event from *found_list*
End if
End if
If event is a regulatory event and *escore*>0.3
Check its argument (protein) for cross-reference
If found
Replace current protein with found event
End if
End if
End for

3 Results and discussion

Table 3 shows the latest results of our system obtained from the online evaluation system (the official evaluation results are 38.19%). The results show that our method performs well on simple and binding events with an F-score of 63.03%. It outperforms previously proposed rule-based systems on these event types despite the fact that part of the test set consists of full text sentences. In addition, our system adapts two syntactic patterns which were previously developed for PPIs extraction. This means that the application of syntactic information is still relevant to extract biological events. In other words, there are some properties these extraction tasks share. However, the performance

[3] A shared trigger is a trigger that appears in more than one group, see section 2.1.

significantly decreases on regulatory events with an F-score of 26.61%.

Analyzing the performance of our system on regulatory events reveals that in most of false positive cases, the errors are caused by not resolving reference events properly. These errors can be reduced if we have a better implementation of the post-processing phase. Another source of errors is that the proposed method did not take into account the dependency among events. For example, most transcription events occurred when the regulatory events occurred (more than 50% cases). If association rules are applied here then the precision of both event types will increase.

Event Class	Recall	Precision	Fscore
Gene_expression	67.27	75.82	71.29
Transcription	46.55	79.41	58.70
Protein_catabolism	40.00	85.71	54.55
Phosphorylation	74.05	80.59	77.18
Localization	44.50	81.73	57.63
Binding	35.23	51.18	41.74
EVT-TOTAL	56.17	71.80	63.03
Regulation	19.22	27.11	22.49
Positive_regulation	22.52	33.89	27.06
Negative_regulation	24.34	33.74	28.28
REG-TOTAL	22.43	32.73	26.61
ALL-TOTAL	38.01	52.06	43.94

Table 3: Evaluation results on test set

To improve the overall performance of the system, there are many issues one should take into account. The first issue is related to the distance or the path length from the joined node between an event trigger and its arguments. By setting a threshold for the distance for each event type we increase the precision of the system. The second issue is related to setting thresholds for the extraction rules (e.g. *t2score, cscore*) which is done by using empirical data. Many interesting challenges remain to be solved, among which are the coreference, anaphora resolution, and cross sentence events. Furthermore, the trade-off between recall and precision needs to be taken into account, setting high thresholds for a dictionary might increase the precision, but could however drop the recall significantly.

4 Conclusion

In this paper we have proposed a novel system which uses syntactic patterns to extract biological events from a text. Our method achieves promising results on simple and binding events. The results also indicate that syntactic patterns for extracting PPIs and biological events share some common properties. Therefore systems developed for extracting PPIs can potentially be used to extract biological events.

Acknowledgements

The authors sincerely thank Dr. Sophia Katrenko and Rick Quax for their useful comments. This work was supported by the European Union through the DynaNets project, EU grant agreement no: 233847, and the Vietnamese Oversea Training Program.

References

S. Ahmed et al. 2009. BioEve: Bio-Molecular Event Extraction from Text Using Semantic Classification and Dependency Parsing. In *Proceedings of the Workshop on BioNLP'09 Shared Task,* pp. 99-102.

G. Móra et al. 2009. Exploring ways beyond the simple supervised learning approach for biological event extraction. In *Proceedings of the Workshop on BioNLP'09 Shared Task,* pp.137-140.

J. Kim et al. 2009. Overview of BioNLP'09 shared task on event extraction. *Proceedings of the Workshop on BioNLP'09 Shared Task,* pp. 1-9.

K. Kaljurand et al. 2009. UZurich in the BioNLP 2009 shared task. In *Proceedings of the Workshop on BioNLP'09 Shared Task*, pp. 28-36.

H. Kilicoglu and S. Bergler. 2009. Syntactic dependency based heuristics for biological event extraction. 2009. In *Proceedings of the Workshop on BioNLP'09 Shared Task*, pp. 119-127.

Q.C. Bui, S. Katrenko, and P.M.A. Sloot. 2011. A hybrid approach to extract protein-protein interactions. *Bioinformatics.* **27**(2), pp. 259-265.

M. Miwa, R. Sætre, J.-D. Kim, and J. Tsujii. 2010. Event Extraction with Complex Event Classification Using Rich Features. *Journal of Bioinformatics and Computational Biology*, **8**, pp. 131-146.

J. Björne, F. Ginter, S. Pyysalo, J. Tsujii, T. Salakoski. 2010. Complex event extraction at PubMed scale. *Bioinformatics*, **26**, pp. i382-390.