A Pattern Approach for Biomedical Event Annotation

Quang Le Minh
Faculty of Information Technology
University of Science
Ho Chi Minh City, Vietnam
leem-inhquang@gmail.com

Son Nguyen Truong
Faculty of Information Technology
University of Science
Ho Chi Minh City, Vietnam
ntson@fit.hcmus.edu.vn

Quoc Ho Bao
Faculty of Information Technology
University of Science
Ho Chi Minh City, Vietnam
hbquoc@fit.hcmus.edu.vn

Abstract

We describe our approach for the GENIA Event Extraction in the Main Task of BioNLP Shared Task 2011. There are two important parts in our method: Event Trigger Annotation and Event Extraction. We use rules and dictionary to annotate event triggers. Event extraction is based on patterns created from dependent graphs. We apply UIMA Framework to support all stages in our system.

1 Introduction

BioNLP Shared Task 2011 has been the latest event following the first attracted event in 2009-2010. We enrolled and submitted the results of Entity Relations Supporting Task and GENIA Event Extraction. In brief, the GENIA task requires the recognition of 9 biological events on genes or gene products described in the biomedical literature. Participants are required to extract and classify 9 kinds of event with appropriate arguments.

First time joining biomedical domain, we aim to learn current problems and approaches in biomedical research. Therefore, we have chosen simple approaches such as rule-based and pattern-based. In the following section, we will explain our work on GENIA Event Extraction Task (GENIA) in details. Finally, we will analyze and discuss results.

2 Our approach

The project uses UIMA Framework[1], an open source framework for analyzing unstructured information, to develop all analysis components. Events bounded in a sentence are 94.4% in training corpus. Consequently, sentences are processed in succession at each step. We divide the whole system into 3 parts: Preprocessing, Event Trigger annotation and Event annotation.

2.1 Preprocessing

At this step, the input documents are converted into objects of the framework. All analysis components will process objects and put results into them. Then we go through natural language processes that include sentence splitting, tokenizing and POS tagging by OpenNLP library. Lastly, the given Protein concepts are annotated.

2.2 Event Trigger annotation

According to our statistics in the training corpus, the percentage of single token trigger is 91.8%. To simplify it, we focus on triggers which span on one token. At this stage, rule-based and dictionary-based approaches are combined.

We choose tokens which are near a protein and have appropriate POS tags. Heuristic rules extracted from training corpus are used to identify candidate triggers. Those rules are, for instance, NN/NNS + of + PROTEIN, VBN + PROTEIN and so on.

Event triggers are diverse in lexical and ambiguous in classification (Björne et al. (2009) and Buyko et al. (2009)). Candidate triggers are classified by a dictionary. The dictionary containing words of triggers with their corresponding classes is built from training corpus. For ambiguous trigger classes, the class that has the highest rate of appearance is chosen.

2.3 Event annotation

Basing on the number of arguments and type of arguments, we categorize 9 event classes into 3 groups. The first group including Gene expression,

[1] Available at http://uima.apache.org/

Proceedings of BioNLP Shared Task 2011 Workshop, pages 149–150,
Portland, Oregon, USA, 24 June, 2011.

Transcription and Protein catabolism has only one Protein as the argument. The second group contains events with Protein and Entity as argument. Phosphorylation, Localization and Binding belong to that group. The third group has the most complex types, i.e. Regulation, Positive regulation and Negative regulation. These events can have other events as their argument.

Our method of event detection is using dependency graph as results of deep syntactic parsing. We prune parse tree and assign concept to nodes. Next, sub-trees which contains only conceptual node as patterns are extracted and represented as string form. We travel breadth-first and write conceptual labels to the string pattern. The pattern list is built from training data.

Firstly, for each sentence contains at least one trigger, we get the parse tree of the sentence. We prune nodes which contain only one child and that child node has zero or one descendant. It reduces the complexity and retains important and general parts of the parse tree.

Secondly, candidate arguments of events are identified by combining Protein, Entity and Event Trigger in that sentence. The number of combination can be huge, so we restrict it by the following conditions. Each combination has at least one Event Trigger with one Protein or Event. The number of argument depends on types of events and is usually less than 5. In addition, the difference of depth on tree between arguments has to be under a threshold.

Thirdly, concepts of arguments in each combination are assigned to parse tree nodes. The assignment bases on the span of argument and content of nodes. The pattern is extracted from the parse tree and examined whether it belongs to the pattern list. In order to increase the precision, we discard patterns having the depth of the tree greater than a threshold. The threshold is chosen by counting on the training corpus.

Finally, we classify events and determine role of arguments for each event. The type of the event is chosen by the type of the trigger of that event. We still simply assign roles of arguments in a fixed order of arguments.

3 Results and conclusions

Our fully official result in GENIA main task is described in Table 1. The F-score is only 14,75% and we were ranked 13th among 14 participants. It reflects many shortcomings in our system. We obtain a lot of experience.

In general, the patterns which we built are still generic. Besides, the OpenNLP library still encountered errors when processing documents, thus affected our result. For example, there are some sentences that OpenNLP parsed or tokenized wrongly and raised errors. In the step of Event Trigger annotation, there are a few rules to cover cases. The result of Regulation, Positive regulation and Negative regulation has the lowest result because we only process recursion with simple events.

Approach	recall	precision	f-score
Gene expression	26.45	39.73	31.76
Transcription	16.09	14.58	15.30
Protein catabolism	33.33	50.00	40.00
Phosphorylation	32.43	47.62	38.59
Localization	16.23	27.68	20.46
Binding	4.68	12.92	6.88
Regulation	0.26	1.35	0.44
Positive regulation	2.08	13.04	3.59
Negative regulation	1.40	11.27	2.49
All Total	**10.12**	**27.17**	**14.75**

Table 1: Our final result in GENIA BioNLP'11 Shared Task with approximately span and recursive matching

For future work, we intend to apply hybrid approach. We combine other methods such as machine learning in Event Trigger and Event annotation parts. We consider other NLP library to improve the performance of all steps relating to NLP processing. Rules from domain professions will be added to existent heuristic rules. We will try to add more features to improve the patterns.

References

Ekaterina Buyko, Erik Faessler, Joachim Wermter and Udo Hahn, "Event Extraction from Trimmed Dependency Graphs," in *Proceedings of the Workshop on BioNLP: Shared Task*, 2009, pp. 19-27.

Jari Bjorne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala and Tapio Salakoski, "Extracting Complex Biological Events with Rich Graph-Based Feature Sets," in *Proceedings of the Workshop on BioNLP: Shared Task*, 2009, pp. 10-18.