BioEve: Bio-Molecular Event Extraction from Text Using Semantic Classification and Dependency Parsing

Syed Toufeeq Ahmed, Radhika Nair, Chintan Patel and Hasan Davulcu
School of Computing and Informatics
Arizona State University
Tempe, Arizona
{toufeeq, ranair1, chpatel, hdavulcu}@asu.edu

Abstract

In this paper, we present **BioEve** a fully automated event extraction system for bio-medical text. It first semantically classifies each sentence to the class type of the event mentioned in the sentence, and then using high coverage hand-crafted rules, it extracts the participants of that event. We participated in Task 1 of BioNLP 2009 Shared task, and the final evaluation results are described here. Our experimentation with different approaches to classify a sentence to bio-interaction classes are also shared.

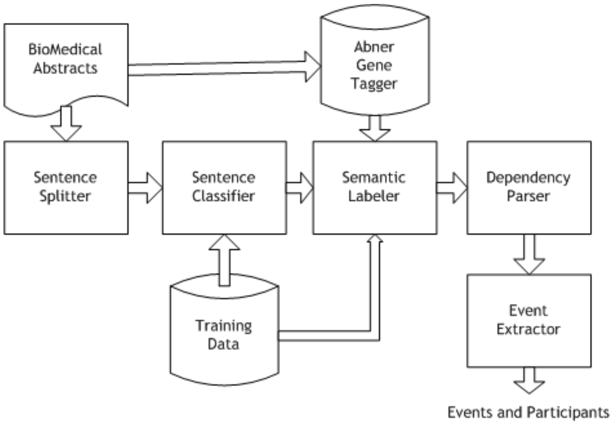

Figure 1: **BioEve** System Architecture

1 Introduction

Human genome sequencing marked beginning of the era of large-scale genomics and proteomics, which in turn led to large amount of information. Lots of that exists (or generated) as unstructured text of published literature. The first step towards extracting event information, in biomedical domain, is to recognize the names of proteins (Fukuda et al., 1998; Blaschke et al., 1999), genes, drugs and other molecules. The next step is to recognize relationship between such entities (Blaschke and Valencia, 2002; Ono et al., 2001; Fundel et al., 2007) and then to recognize the bio-molecular interaction events with these entities as participants (Yakushiji et al., 2001; Tateisi et al., 2004). The BIONLP'09 shared task involved recognition of bio-molecular events, which appear in the GENIA corpus. We mainly focused on task 1, which was detection of an event and its participants.

The rest of the paper is organized as follows. In Section 2 we describe BioEve system, sentence level classification and event extraction using dependency parse tree of the sentence. Sections 3 describes experiments with classification approaches and evaluation results for shared task 1. Section 4 concludes the paper.

2 BioEve: Bio-Molecular Event Extractor

BioEve architecture is shown in Figure 1. First the biomedical abstracts are split into sentences, before being sent to sentence level classifier. We used Näive Bayes Classifier to classify sentences into different event class types. Classification at sentence level is a difficult task, as sentences have lesser information as compared to the whole document. To help event extraction module, each of these sentences are then semantically labeled with additional keywords. We created a dictionary-based

Proceedings of the Workshop on BioNLP: Shared Task, pages 99–102,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

labeler, which included trigger words from training data, along with the corresponding event type. These labeled sentences are parsed using a dependency parser to identify `argument-predicate` roles. For each event class type, we hand crafted high coverage extraction rules, similar to Fundel et al. (2007), to identity all event participants. For BioNLP shared task, the event-participant output was formatted to GENIA format.

2.1 Sentence Level Classification and Semantic Labeling

We used Näive Bayes Classifier from Weka [1] library to classify sentences into different event class types. Classification at sentence level is a difficult task, as sentences have lesser information as compared to the whole document. We tried different approaches for classification : 1) Näive Bayes Classifier using bag-of-words, 2) Näive Bayes Classifier using bag-of-words and parts-of-speech tags and 3) SVM Classifier for Weka library.

BioEve event extraction module depends on class labels for extraction. To help with this task, we needed to improve sentence labeling with correct class type information. For this, we employed dictionary based semantic class labeling by identifying trigger (or interaction) words, which clearly indicate presence of a particular event. We used ABNER [2] gene name recognizer to enrich the sentences with gene mentions.

There have been cases in the training data where the same trigger word is associated with more than one event type. To resolve such cases, the trigger words were mapped to the most likely event type based on their occurrence count in the training data. We labeled trigger words in each sentence with their most likely event type. These tagged words served as a starting point for the extraction of event participants. This was done to speed-up the extraction process, as event extraction module now only needs to focus on the parts of the sentences related to these tagged trigger words.

2.2 Event Extraction Using Dependency Parsing

The sentences, after being class labeled and tagged, are parsed using a dependency parser (Stanford parser[3]) to identify `argument-predicate` roles. Words in the sentence and the relationships between these words form the dependency parse tree of the sentence. For our system, we used typed-dependency representation output format from Stanford parser which is a simple tuple, `reln(gov, dep)`, where `reln` is the dependency relation, `gov` is the governor word and `dep` is the dependent word. Consider the following example sentence:

```
We investigated whether PU.1 binds
and activates the M-CSF receptor
promoter.
```

After this sentence is class labeled and tagged:

```
We investigated whether
T7 binds/BINDING and
activates/POSITIVE_REGULATION the
T8 promoter.
```

The tagged sentence is parsed to obtain dependency relations as shown below:

```
nsubj(investigated-2, We-1)
complm(binds-5, whether-3)
nsubj(binds-5, T7-4)
ccomp(investigated-2, binds-5)
conj_and(binds-5, activates-7)
det(promoter-10, the-8)
nn(promoter-10, T8-9)
dobj(binds-5, promoter-10)
```

This sentence mentions two separate events, *binding* and *positive regulation*. Let's consider the extracting the event *binding* and its participants. Figure 2 shows the parse tree representation and the part of the tree that needs to be identified for extracting event *binding*.

For each event class type, we carefully hand crafted rules, keeping theme of the event, number of participants, and their interactions into consideration. Table 1 lists these extraction rules. In an extraction rule, `T` represents the occurrence of protein in sentence. If multiple proteins are involved, then subscripts, $\mathtt{T}_n$, are used to represent this. The rule

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] http://pages.cs.wisc.edu/ bsettles/abner/

[3] http://nlp.stanford.edu/software/lex-parser.shtml

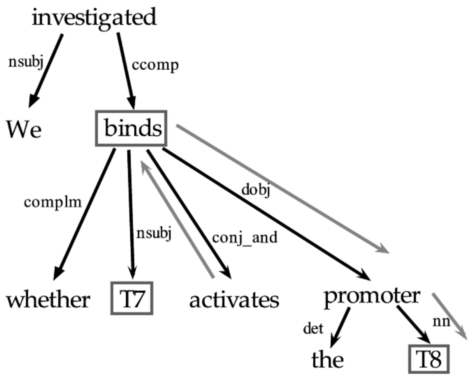

Figure 2: Dependency Parse tree, and event ”binding” and its participants are shown.

is triggered when it matches I (for an *interaction word*, or *trigger word*) in the sentence. Some dependency relations and rule predicates are explained below:

- obj(verb/I, T) :- The matching protein is a direct object of the interaction word
- prep(I, T) :- The matching protein is connected to its interaction word by a preposition
- T_1 (I) T_2 : − The interaction word occurs in between the two matching interacting proteins
- conj(T_1, T_2) The two matching proteins are be connected to each other using conjugates such as *’and’*
- ConnectedRule :- The interaction word and the matching protein should be directly connected with a single edge (dependency relation)
- NearestRule :- The interaction word and the matching protein should be connected to each other, directly or indirectly within 5 edge hops, in either direction

Algorithm 1 shows the steps to extract event participants using the rules given in Table 1.

3 Experiments and Evaluations

BioEve shared task evaluation results for Task 1 are shown in Table 2. Event extraction for classes *gene-expression, protein-catabolism and phosphorylation* performed better comparatively, where as, for classes *transcription, regulation, positive-regulation and negative-regulation*, it was below par. The reason noticed (in training examples) was that, most of the true example sentences of *positive-regulation or negative-regulation* class type were mis-classified as either *phosphorylation or gene-expression*. This calls for further improvement of sentence classifier accuracy. Experiments with different approaches for sentence level classification are shown in Table 3. Classifiers were trained on training data and tested on development data. Interestingly, simple Näive Bayes Classifier (NBC) (using just bag-of-words (BOW)) showed better results (up to 10% better) compared to other approaches, even SVM classifier.

```
Input: Abstract tagged with interaction words
       and class labels
Output: Bio Events with interaction words and
        the participants
foreach abstract do Iterate over each abstract
    foreach sentence in current abstract do
        retrieve all the interaction words in
        current sentence;
        sort them according to precedence of the
        event class type;
        foreach interaction word in the sentence
        do
            extract the participants by matching
            the corresponding event's rule to the
            sentence's dependency parse;
        end
    end
end
```

Algorithm 1: BioEve Event Extraction algorithm

4 Conclusions

In this paper, **BioEve**’s Task 1 evaluation results were described, with additional results from different approaches experimented to semantically classify a sentence to the event type. Event extraction performed better for some categories, but clearly needs re-compiling extraction rules for some. Where as classification results showed simple Näive Bayes Classifier performing better than other approaches.

Event Class	Extraction Rules
Positive Regulation	a) obj(verb/I, T) b) prep(I, T) c) ConnectedRule d) NearestRule
Regulation	a) prep(I, T) b) ConnectedRule c) NearestRule
Phosphorylation	a) prep(I, T) b) T (connecting-word) I c) ConnectedRule d) NearestRule
Gene Expression	a) ConnectedRule b) NearestRule
Transcription	a) prep(I, T) b) T (connecting-word) I c) ConnectedRule d) NearestRule

Event Class	Extraction Rules
Negative Regulation	a) obj(verb/I, T) b) prep(I, T) c) ConnectedRule d) NearestRule
Binding	a) T_1 (I) T_2 b) prep(I, T_1); prep(T_1, T_2) c) prep(I, T_1); conj(T_1, T_2) d) obj(verb/I, T) e) prep(I, T) f) ConnectedRule g) NearestRule
Protein Catabolism	a) prep(I, T) b) ConnectedRule c) NearestRule
Localization	a) prep(I, T) b) ConnectedRule c) NearestRule

Table 1: Extraction rules for each class type. Rules are fired in the order they are listed for each class.

Approach	recall	precision	f-score
Localization	27.59	33.57	30.28
Binding	16.71	30.53	21.60
Gene-expression	**44.04**	**39.55**	**41.68**
Transcription	10.95	11.28	11.11
Prot-catabolism	**57.14**	27.59	37.21
Phosphorylation	**50.37**	**63.55**	**56.20**
Regulation	9.28	5.18	6.65
Pos-regulation	10.48	7.34	8.63
Neg-regulation	12.93	10.19	11.40
All Total	**21.81**	**18.21**	**19.85**

Table 2: BioNLP Shared Task Evaluation: Task 1 Results using approximate span matching.

Sentence Classifier	Correct	Incorrect
NBC(BOW)	**60.45%**	**39.54%**
NBC(BOW+POS)	43.12%	56.87%
SVM	50.14%	49.85%

Table 3: Sentence Classifier results for different approaches: 1) Näive Bayes Classifier (NBC) (using bag-of-words (BOW)), 2) Näive Bayes Classifier(using BOW + Parts-of-speech(POS) tags) and 3) SVM Classifier. Total number of instances =**708**.

References

Christian Blaschke and Alfonso Valencia. 2002. The frame-based module of the suiseki information extraction system. *IEEE Intelligent Systems*, 17(2):14–20.

C. Blaschke, MA. Andrade, C. Ouzounis, and A. Valencia. 1999. Automatic extraction of biological information from scientific text: protein-protein interaction. In *Proceedings of the AAAI conference on Intelligent Systems in Molecular Biology*, pages 60–7. AAAI.

K. Fukuda, A. Tamura, T. Tsunoda, and T. Takagi. 1998. Toward information extraction: identifying protein names from biological papers. In *Pac Symp Biocomput*, volume 707, page 18.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Toshihide Ono, Haretsugu Hishigaki, Akira Tanigami, and Toshihisa Takagi. 2001. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161.

Y. Tateisi, T. Ohta, and J. Tsujii. 2004. Annotation of predicate-argument structure of molecular biology text. In *JCNLP-04 workshop on Beyond Shallow Analyses*.

Akane Yakushiji, Yuka Tateisi, Yusuke Miyao, and Jun ichi Tsujii. 2001. Event extraction from biomedical papers using a full parser. In *Pac. Symp. Biocomput*, pages 408–419.