Supervised Classification for Extracting Biomedical Events

Arzucan Özgür
Department of EECS
University of Michigan
Ann Arbor, MI 48109, USA
ozgur@umich.edu

Dragomir R. Radev
Department of EECS and
School of Information
University of Michigan
Ann Arbor, MI 48109, USA
radev@umich.edu

Abstract

We introduce a supervised approach for extracting bio-molecular events by using linguistic features that represent the contexts of the candidate event triggers and participants. We use Support Vector Machines as our learning algorithm and train separate models for event types that are described with a single theme participant, multiple theme participants, or a theme and a cause participant. We perform experiments with linear kernel and edit-distance based kernel and report our results on the BioNLP'09 Shared Task test data set.

1 Introduction

Most previous work on biomedical information extraction focuses on identifying relationships among biomedical entities (e.g. protein-protein interactions). Unlike relationships, which are in general characterized with a pair of entities, events can be characterized with event types and multiple entities in varying roles. The BioNLP'09 Shared Task addresses the extraction of bio-molecular events from the biomedical literature (Kim et al., 2009). We participated in the "Event Detection and Characterization" task (Task 1). The goal was to recognize the events concerning the given proteins by detecting the event triggers, determining the event types, and identifying the event participants.

In this study, we approach the problem as a supervised classification task. We group the event types into three general classes based on the number and types of participants that they involve. The first class includes the event types that are described with a single theme participant. The second class includes the event types that are described with one or more theme participants. The third class includes the events that are described with a theme and/or a cause participant. We learn support vector machine (SVM) models for each class of events to classify each candidate event trigger/participant pair as a real trigger/participant pair or not. We use various types of linguistic features such as lexical, positional, and dependency relation features that represent the contexts of the candidate trigger/participant pairs. The results that we submitted to the shared task were based on using a linear kernel function. In this paper, we also report our results based on using an edit-distance based kernel defined on the shortest dependency relation type paths between a candidate trigger/participant pair.

2 System Description

2.1 Event Type Classes

We grouped the nine event types targeted at the BioNLP'09 Shared Task into three general event classes based on the number and types of participants that they involve.

Class 1 Events: Events that involve a single theme participant (Gene expression, Transcription, Protein catabolism, Localization, and Phosphorylation event types).

Class 2 Events: Events that can involve one or more theme participants (Binding event type).

Class 3 Events: Events that can be described with a theme and/or a cause participant (Regulation, Positive regulation, and Negative regulation event types). Unlike Class 1

Proceedings of the Workshop on BioNLP: Shared Task, pages 111–114,
Boulder, Colorado, June 2009.

and Class 2 events, where the participants are proteins, the participants of Class 3 events can be proteins or events.

Since the event types in each class are similar to each other based on the number and roles of participants that they involve and different from the event types in the other classes, we learned separate classification models for each class. We formulated the classification task as the classification of trigger/participant pairs. We extracted positive and negative training instances (trigger/participant pairs) from the training data for each class of events. We considered only the pairs that appear in the same sentence. We used the tokenized and sentence split abstracts provided by the shared task organizers[1]. Consider the sentence *"The phosphorylation of TRAF2 inhibits binding to the CD40 cytoplasmic domain"*. This sentence describes the following three events:

1. Event1: Type: Phosphorylation Trigger: phosphorylation Theme: TRAF2
2. Event2: Type: Binding Trigger: binding Theme1: TRAF2 Theme2: CD40
3. Event3: Type: Negative regulation Trigger: inhibits Theme: Event2 Cause: Event1

Event1 belongs to Class 1. The trigger/participant pair (phosphorylation, TRAF2) is a positive instance for Class 1. Event2 belongs to Class 2. It has two theme participants. The instances for Class 2 events are created by decomposing the events into trigger/theme pairs. The two positive instances extracted from the decomposition of Event2 are (binding, TRAF2) and (binding, CD40). Event3 belongs to Class 3. It consists of two semantically different participants, namely a theme and a cause. We trained two separate models for Class 3 events, i.e., one model to classify the themes and another model to classify the causes. Another distinguishing characteristic of Class 3 events is that a participant of an event can be a protein or an event. We represent the participants that are events with their corresponding event triggers. We decompose Event3 into its theme and cause and represent its cause Event1 with its trigger word "phosphorylation" and its theme Event2 with its trigger word "binding". As a result, (inhibits, binding) and (inhibits, phosphorylation) are included as positive instances to the Class 3 theme and Class 3 cause training sets, respectively. Negative instances for Class 1 and Class 2 are created by including all the trigger/protein pairs which are not among the positive instances of that class. Negative instances for Class 3 theme and Class 3 cause are created by including all the trigger/protein and trigger1/trigger2 pairs which are not among the positive instances of that class. For example, (phosphorylation, CD40) is a negative instance for Class 1 and (inhibits, TRAF2) is a negative instance for Class 3 theme and Class 3 cause.

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/tools.html

2.2 Feature Extraction

2.2.1 Lexical and Part-of-Speech Features

We used the candidate trigger and its part-of-speech, which was obtained by using the Stanford Parser, as features, based on our observation that different candidate triggers might have different likelihoods of being a real trigger for a certain event. For example, *"transcription"* is a trigger for the Transcription event 277 times in the training set and has not been used as a trigger for other types of events. On the other hand, *"concentration"* is used only once as a trigger for a Transcription event and three times as a trigger for Regulation events.

2.2.2 Positional Features

We used two features to represent the relative position of the participant with regard to the trigger in the sentence. The first feature has two values, namely "before" (the participant appears before the trigger) or "after" (the participant appears after the trigger). The second feature encodes the distance between the trigger and the participant. Distance is measured as the number of tokens between the trigger and the participant. Our intuition is that, if a candidate trigger and participant are far away from each other, it is less likely that they characterize an event.

2.2.3 Dependency Relation Features

A dependency parse tree captures the semantic predicate-argument dependencies among the words of a sentence. Dependency tree paths between protein pairs have successfully been used to identify

protein interactions (Bunescu and Mooney, 2007; Erkan et al., 2007). In this paper, we use the dependency paths to extract events. For a given trigger/participant pair, we extract the shortest path from the trigger to the participant, from the dependency parse of the sentence. We use the *McClosky-Charniak* parses which are converted to the Stanford Typed Dependencies format and provided to the participants by the shared task organizers. Previous approaches use both the words and the dependency relation types to represent the paths (Bunescu and Mooney, 2007; Erkan et al., 2007). Consider the dependency tree in Figure 1. The path from "phosphorylation" to "CD40" is "nsubj inhibits acomp binding prep_to domain num". Due to the large number of possible words, using the words on the paths might lead to data sparsity problems and to poor generalization. Suppose we have a sentence with similar semantics, where the synonym word "prevents" is used instead of "inhibits". If we use the words on the path to represent the path feature, we end up with two different paths for the two sentences that have similar semantics. Therefore, in this study we use only the dependency relation types among the words to represent the paths. For example, the path feature extracted for the (phosphorylation, CD40) negative trigger/participant pair is "nsubj acomp prep_to num" and the path feature extracted for the (phosphorylation, TRAF2) positive trigger/participant pair is "prep_of".

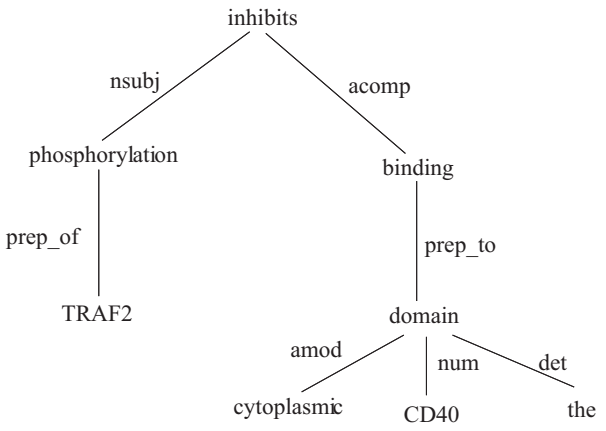

Figure 1: The dependency tree of the sentence "*The phosphorylation of TRAF2 inhibits binding to the CD40 cytoplasmic domain.*"

2.3 Classification

We used the SVM^{light} library (Joachims, 1999) with two different kernel functions and feature sets for learning the classification models. Our first approach is based on using linear SVM with the features described in Section 2.2. In this approach the path feature is used as a nominal feature. Our second approach is based on integrating to SVM a kernel function based on the word-based edit distance between the dependency relation paths, where each dependency relation type on the path is treated as a word. For example, the word-based edit distance between the paths "prep_of" and "prep_of prep_with" is 1, since 1 insertion operation (i.e., inserting "prep_with" to the first path) is sufficient to transform the first path to the second one. The edit-distance based similarity between two paths p_i and p_j and the corresponding kernel function are defined as follows (Erkan et al., 2007).

$$edit\_sim(p_i, p_j) = e^{-\gamma(edit\_distance(p_i, p_j))} \quad (1)$$

3 Experimental Results

The data provided for the shared task is prepared from the GENIA corpus (Kim et al., 2008). We used the training and the development sets for training.

The candidate triggers are detected by using a dictionary based approach, where the dictionary is extracted from the training set. We filtered out the noisy trigger candidates such as "with", "+", ":", and "-", which are rarely used as real triggers and commonly used in other contexts. The candidate trigger/participant pairs are classified by using the classifiers learned for Class 1, Class 2, and/or Class 3 depending on whether the candidate trigger matched one of the triggers in these classes. The SVM score is used to disambiguate the event types, if a candidate trigger matches a trigger in more than one of the event classes. A trigger which is ambiguous among the event types in the same class is assigned to the event type for which it is most frequently used as a trigger.

The results that we submitted to the shared task were obtained by using the linear SVM approach with the set of features described in Section 2.2. After submitting the results, we noticed that we made an error in pre-processing the data set. While aligning the provided dependency parses with the

sentence, we incorrectly assumed that all the sentences had dependency parses and ended up using the wrong dependency parses for most of the sentences. The overall performance scores for our official submission are 30.42% recall, 14.11% precision, and 19.28% F-measure. The results obtained after correcting the error are reported in Table 1. Correcting the error significantly improved the performance of the system. Table 2 shows the results obtained by using SVM with dependency path edit kernel. The two SVM models achieve similar performances. The performance for the regulation events is considerably lower, since errors in identifying the events are carried to identifying the event participants of a regulation event. The performances for the events which have multiple participants, i.e., binding and regulation events, are lower compared to the events with a single participant. The performance is higher when computed by decomposing the events (49.00 and 31.82 F-measure for binding and regulation events, respectively). This suggests that even when participants of events are identified correctly, there is significant amount of error in composing the events.

Event Type	Recall	Precision	F-measure
Localization	41.95	60.83	49.66
Binding	31.41	34.94	33.08
Gene_expression	61.36	69.00	64.96
Transcription	37.23	30.72	33.66
Protein_catabolism	64.29	64.29	64.29
Phosphorylation	68.15	80.70	73.90
Event Total	50.82	56.80	53.64
Regulation	15.12	19.82	17.15
Positive_regulation	24.21	33.33	28.05
Negative_regulation	21.64	32.93	26.11
Regulation Total	22.02	30.72	25.65
All Total	35.86	44.69	39.79

Table 1: Approximate span & recursive matching results using linear SVM with the set of features described in Section 2.2 (after correcting the error in pre-processing the data set).

4 Conclusion

We described a supervised approach to extract biomolecular events. We grouped the event types into three general classes based on the number and types of participants that they can involve and learned separate SVM models for each class. We used various types of linguistic features that represent the context of the candidate event trigger/participant pairs. We achieved an F-measure of 39.83% on the shared task test data. Error analysis suggests that improving the approach of event composition for types of events with multiple participants and improving the strategy for detecting and disambiguating triggers can enhance the performance of the system.

Event Type	Recall	Precision	F-measure
Localization	49.43	64.18	55.84
Binding	31.70	35.03	33.28
Gene_expression	66.34	69.72	67.99
Transcription	39.42	25.59	31.03
Protein_catabolism	78.57	73.33	75.86
Phosphorylation	76.30	80.47	78.33
Event Total	55.13	56.62	55.86
Regulation	17.87	16.46	17.13
Positive_regulation	26.45	26.03	26.24
Negative_regulation	25.33	32.54	28.49
Regulation Total	24.68	25.34	25.01
All Total	39.31	40.37	39.83

Table 2: Approximate span & recursive matching results using SVM with dependency relation path edit kernel.

Acknowledgments

This work was supported in part by the NIH Grant U54 DA021519.

References

R. C. Bunescu and R. J. Mooney, 2007. *Text Mining and Natural Language Processing*, Chapter Extracting Relations from Text: From Word Sequences to Dependency Paths, pages 29–44, Springer.

Güneş Erkan, Arzucan Özgür, and Dragomir R. Radev. 2007. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of EMNLP*, pages 228–237.

T. Joachims, 1999. *Advances in Kernel Methods-Support Vector Learning*, Chapter Making Large-Scale SVM Learning Practical. MIT-Press.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1).

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 Natural Language Processing in Biomedicine (BioNLP) NAACL 2009 Workshop*. To appear.