Identification of Genia Events using Multiple Classifiers

Roland Roller and **Mark Stevenson**
Department of Computer Science,
University of Sheffield
Regent Court, 211 Portobello
Sheffield, S1 4DP
United Kingdom
{R.Roller, M.Stevenson}@dcs.shef.ac.uk

Abstract

We describe our system to extract genia events that was developed for the BioNLP 2013 Shared Task. Our system uses a supervised information extraction platform based on Support Vector Machines (SVM) and separates the process of event classification into multiple stages. For each event type the SVM parameters are adjusted and feature selection carried out. We find that this optimisation improves the performance of our approach. Overall our system achieved the highest precision score of all systems and was ranked 6th of 10 participating systems on F-measure (strict matching).

1 Introduction

The BioNLP 2013 Shared Task focuses on information extraction in the biomedical domain and comprises of a range of extraction tasks. Our system was developed to participate within the Genia Event Extraction task (GE), which focuses on the detection of gene events and their regulation. The task considers 13 different types of events which can be divided into four groups: simple events, bindings, protein modifications and regulations. All events consist of a core event, which contains a trigger word and a theme. With the exception of regulation events, the theme always refer to a protein. A regulation event theme can either refer to a protein or to another event. Binding events can include up to two proteins as themes. In addition to the core event, events may include additional arguments such as ‘cause’ or ‘to location’.

Figure 1 shows examples of events from the BioNLP 2013 corpus. More details about the Genia Event task can be found in Kim et al. (2011).

Previous editions of the BioNLP Shared Task took place in 2009 (Kim et al., 2009) and 2011 (Kim et al., 2011). Promising approaches in the most recent competition were event parsing (McClosky et al., 2011) and dual decomposition models (Riedel and McCallum, 2011). The winner of the GE task 2011, FAUST (Riedel et al., 2011), combined these two approaches by using result from the event parser as an additional input feature for the dual decomposition.

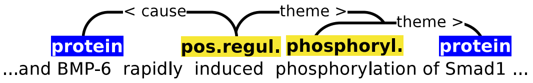

Figure 1: Two events from the BioNLP 2013 GE task: a phosphorylation event consisting of a trigger and a protein and a positive-regulation event consisting of a trigger, a theme referring to an event and a cause argument.

The UTurku system of Björne et al. (2009) was the winner of the GE task in 2009. The system was based on a pipeline containing three main stages: trigger detection, argument detection and post-processing. Björne and Salakoski (2011) improved the performance of this system for BioNLP 2011, but was outperformed by FAUST.

Our approach to the BioNLP Shared Task relies on separating the process of event classification into multiple stages and creates separate classifiers for each event type. Our system begins by pre-processing the input text, followed by multiple classification stages and a post-processing stage. The pre-processing applies tokenization, sentence splitting and dictionary-based trigger detection, similar to Bui and Sloot (2011). Classification is based on a Support Vector Machine (SVM) and uses three main stages: trigger-protein detection, trigger-event detection and event-cause detection. Post-processing is a combination of classification and rule-based approaches. We train a separate classifier for each event type, rather that relying on a single classifier to recognise trigger-theme rela-

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 125–129,
Sofia, Bulgaria, August 9 2013. ©2013 Association for Computational Linguistics

tionships for all event types. In addition, we also optimise the SVM's parameters and apply feature selection for each event type.

Our system participated in subtask 1 of the GE task, which involves the recognition of core events, including identification of their 'cause'.

The remainder of this paper describes our system in detail (Section 2), presents results from the Genia Event Extraction task (Section 3) and draws the conclusions of this work (Section 4).

2 System Description

2.1 Preprocessing

Our system begins by preprocessing the input text, by applying the sentence splitter and biomedical named entity tagger from LingPipe[1]. The sentence splitter is trained on the MEDLINE data set. The text is then tokenised. Tokens containing punctuation marks are split, as are tokens containing a protein or suffixes which could be utilised as a trigger word. For instance the term 'Foxp3-expression' will be split into 'Foxp3 - expression', since 'Foxp3' is as a protein and 'expression' a suffix often used as trigger word. The tokens are then stemmed using the Porter Stemmer from the NLTK[2] toolkit. The Stanford Parser[3] is used to extract part-of-speech tags, syntax trees and dependency trees.

2.1.1 Trigger Detection

The names of proteins in the text are provided in the GE task, however the trigger words that form part of the relation have to be identified. Our system uses a dictionary-based approach to trigger detection. The advantage of this approach is that it is easy to implement and allows us to easily identify as many potential trigger words as possible. However, it will also match many words which are not true triggers. We rely on the classification stage later in our approach to identify the true trigger words.

A training corpus was created by combining the training data from the 2013 Shared Task with all of the data from the 2011 task. All words that are used as a trigger in this corpus are extracted and stored in a set of dictionaries. Separate dictionaries are created for different event types (e.g. localization, binding). Each type has its own dictionary, with the exception of protein modification events (protein modification, phosphorylation, ubiquitination, acetylation, deacetylation). The corpus did not contain enough examples of trigger terms for these events and consequently they are combined into a single dictionary. The words in the dictionaries are stemmed and sorted by their frequency. Irrelevant words (such as punctuations) are filtered out.

Trigger detection is carried out by matching the text against each of the trigger dictionaries, starting with the trigger words with the highest frequency. A word may be annotated as a trigger word by different dictionaries. If a word is annotated as a trigger word for a specific event then it may not be annotated as being part of another trigger word from the same dictionary. This restriction prevents the generation of overlapping trigger words for the same event as well as preventing too many words being identified as potential triggers.

2.2 Classification

Classification of relations is based on SVM with a polynomial kernel, using LibSVM (Chang and Lin, 2011), and is carried out in three stages. The first covers the core event, which consists of a trigger and a theme referring to a protein. The second takes all classified events and tries to detect regulation events consisting of a trigger and a theme that refers to one of these events (see positive-regulation event in figure 1). In addition to a trigger and theme, regulation and protein modification events may also include a cause argument. The third stage is responsible for identifying this additional argument for events detected in the previous two stages.

Classification in each stage is always between pairs of object: trigger-protein (stage 1), trigger-event (stage 2), event-protein (stage 3) or event-event (stage 3). At each stage the role of the classifier is to determine whether there is in fact a relation between a given pair of objects. This approach is unable to identify binding events involving two themes. These are identified in a post-processing step (see Section 2.3) which considers binding events involving the same trigger word and decides whether they should be merged or not.

2.2.1 Feature Set

The classification process uses a wide range of features constructed from words, stemmed words, part of speech tags, NE tags and syntactic analysis.

[1] http://alias-i.com/lingpipe/index.html

[2] http://nltk.org/

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

Object Features: The classification process always considers a pair of objects (e.g. trigger-protein, trigger-event, event-protein). Object features are derived from the tokens (words, stemmed words etc.) which form the objects. We consider the head of this object, extracted from the dependency tree, as a feature and all other tokens within that object as bag of word features. We also consider the local context of each object and include the three words preceding and following the objects as features.

Sentence Features: The tokens between the two objects are also used to form features. A bag of word is formed from the tokens between the features and, in addition, the complete sequence of tokens is also used as a feature. Different sentence features are formed from the words, stemmed words, part of speech tags and NE tags .

Syntactic Features: A range of features are extracted from the dependency and phrase-structure trees generated for each sentence. These features are formed from the paths between the the objects within dependency tree, collapsed dependency tree and phrase-structure tree. The paths are formed from tokens, stemmed tokens etc.

The features are organised into 57 groups for use in the feature selection process described later. For example all of the features relating to the bag of words between the two objects in the dependency tree are treated as a single group, as are all of the features related to the POS tags in the three word range around one of the objects.

2.2.2 Generation of Training and Test Data

Using the training data, a set of positive and negative examples were generated to train our classifiers. Pairs of entities which occur in a specific relation in the training data are used to generate positive examples and all other pairs used to generate negative ones. Since we do not attempt to resolve coreference, we only consider pairs of entities that occur within the same sentence.

Due to the fact that we run a dictionary-based trigger detection on a stemmed corpus we might cover many trigger words, but unfortunately also many false ones. To handle this situation our classifier should learn whether a word serves as a trigger of an event or not. To generate sufficient negative examples we also run the trigger detection on the training data set, which already contains the right trigger words.

2.2.3 Classifier optimisation

Two optimisation steps were applied to the relation classifiers and found to improve their performance.

SVM bias adjustment: The ratio of positive and negative examples differs in the training data generated for each relation. For instance the data for the protein catabolism event contains 156 positive examples and 643 negatives ones while the gene expression event has 3617 positive but 34544 negative examples. To identify the best configuration for two SVM parameters (cost and gamma), we ran a grid search for each classification step using 5-fold cross validation on the training set.

Feature Selection: We also perform feature selection for each event type. We remove each feature in turn and carry out 5-fold cross validation on the training data to identify whether the F-measure improves. If improvement is found then the feature that leads to the largest increase in F-measure is removed from the feature set for that event type and the process repeated. The process is continued until no improvement in F-measure is observed when any of the features are removed. The set of features which remain are used as the final set for the classifier.

The feature selection shows the more positive training examples we have for an event type the fewer features are removed. For example, gene expression events have the highest amount of positive examples (3617) and achieve the best F-measure score without removing any feature. On the other hand, there are just 156 training examples for protein catabolism events and the best results are obtained when 39 features are removed. On average we remove around 14 features for each event classifier. We observed that sentence features and those derived from the local context of the object are those which are removed most often.

2.3 Post-Processing

The output from the classification stage is post-processed in order to reduce errors. Two stages of post-processing are applied: one of which is based on a classifier and another which is rule based.

Binding Re-Ordering: As already mentioned in Section 2.2, our classification is only capable of detecting single trigger-protein bindings. However if two binding events share the same trigger, they could be merged into a single binding containing two themes. A classifier is trained to decide whether to merge pairs of binding events. The classifier is provided with the two themes that share a trigger word and is constructed in the same way as the classifiers that were used for relations. We utilise the same feature set as in the other classification steps and run a grid search to adjust the SVM parameter to decide whether to merge two bindings or not.

Rule-Based Post-Processing: The second stage of post-processing considers all the events detected within a sentence and applies a set of manually created rules designed to select the most likely. Some of the most important rules include:

- Assume that the classifier has identified both a simple event (e_1) and regulation event (e_2) using the same trigger word and theme. If another event uses a different trigger word with e_1 as its theme then e_2 is removed.

- If transcription and gene expression events are identified which use the same trigger and theme then the gene expression event is removed. This situation occurs since transcription is a type of a gene expression and the classifiers applied in Section 2.2 may identify both types.

- Assume there are two events (e_1 and e_2) of the same type (e.g. binding) that use the same trigger word but refer to different proteins. If the theme of a regulation event refers to e_1 then a new regulation event referring to e_2 is introduced.

3 Results

Our approach achieved the highest precision score (63.00) in the formal evaluation in terms of strict matching in the GE task 1. The next highest precision scores were achieved by BioSEM (60.67) and NCBI (56.72). We believe that the classifier optimisation (Section 2.2.3) for each event and the use of manually created post-processing rules (Section 2.3) contributed to the high precision score. Our system was ranked 6th place of 10 in terms of F-measure with a score of 42.06.

Table 1 presents detailed results of our system for the GE task. Our approach leads to high precision scores for many of the event types with a precision of 79.23 for all simple events and 92.68 for protein modifications. Our system's performance is lower for regulation events than other types with a precision of 52.69. Unlike other types of events, the theme of a regulation event may refer to another event. The detection of regulation events can therefore be affected by errors in the detection of simple events.

Results of our system are closer to the best reported results when strict matching is used as the evaluation metric. In this case the F-measure is 6.86 lower than the winning system (BioSEM). However, when the approximate span & recursive matching metric is used the results of our system are 8.74 lower than the best result, which is achieved by the EVEX system.

Event Class	**Recall**	**Prec.**	**Fscore**
Gene_expression	62.20	85.37	71.96
Transcription	33.66	45.33	38.64
Protein_catabolism	57.14	53.33	55.17
Localization	23.23	85.19	36.51
SIMPLE ALL	54.02	79.23	64.24
Binding	31.53	46.88	37.70
Phosphorylation	47.50	92.68	62.81
PROT-MOD ALL	39.79	92.68	55.68
Regulation	11.46	42.86	18.08
Positive_regulation	23.72	53.60	32.88
Negative_regulation	20.91	54.19	30.18
REG. ALL	21.14	52.69	30.18
EVENT TOTAL	31.57	63.00	42.06

Table 1: Evaluation Results (strict matching)

4 Conclusion

Our approach to the BioNLP GE task 1 was to create a separate SVM-based classifier for each event type. We adjusted the SVM parameters and applied feature selection for each classifier. Our system post-processed the outputs from these classifiers using a further classifier (to decide whether events should be merged) and manually created rules (to select between conflicting events). Results show that our approach achieves the highest precision of all systems and was ranked 6th in terms of F-measure when strict matching is used.

In the future we would like to improve the recall of our approach and also aim to explore the use of a wider range of features. We would also like to experiment with post-processing based on a classifier and compare performance with the manually created rules currently used.

References

Jari Björne and Tapio Salakoski. 2011. Generalizing biomedical event extraction. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 183–191, Portland, Oregon, USA, June. Association for Computational Linguistics.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 10–18, Boulder, Colorado, June. Association for Computational Linguistics.

Quoc-Chinh Bui and Peter. M.A. Sloot. 2011. Extracting biological events from text using simple syntactic patterns. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 143–146, Portland, Oregon, USA, June. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

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 the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado, June. Association for Computational Linguistics.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. Overview of genia event task in bionlp shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 7–15, Portland, Oregon, USA, June. Association for Computational Linguistics.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing for bionlp 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 41–45, Portland, Oregon, USA, June. Association for Computational Linguistics.

Sebastian Riedel and Andrew McCallum. 2011. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 46–50, Portland, Oregon, USA, June. Association for Computational Linguistics.

Sebastian Riedel, David McClosky, Mihai Surdeanu, Andrew McCallum, and Christopher D. Manning. 2011. Model combination for event extraction in bionlp 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 51–55, Portland, Oregon, USA, June. Association for Computational Linguistics.