From Protein-Protein Interaction to Molecular Event Extraction

Rune Sætre[†], **Makoto Miwa**[†], **Kazuhiro Yoshida**[‡] and **Jun'ichi Tsujii**[†]
{rune.saetre,mmiwa,kyoshida,tsujii}@is.s.u-tokyo.ac.jp
[†]Department of Computer Science
[‡]Information Technology Center
University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan

Abstract

This document describes the methods and results for our participation in the BioNLP'09 Shared Task #1 on Event Extraction. It also contains some error analysis and a brief discussion of the results. Previous shared tasks in the BioNLP community have focused on extracting *gene and protein names*, and on finding (direct) protein-protein interactions (PPI). This year's task was slightly different, since the protein names were already manually annotated in the text. The new challenge was to extract biological *events* involving these given *gene and gene products*. We modified a publicly available system (AkanePPI) to apply it to this new, but similar, protein interaction task. AkanePPI has previously achieved state-of-the-art performance on all existing public PPI corpora, and only small changes were needed to achieve competitive results on this event extraction task. Our official result was an F-score of 36.9%, which was ranked as number six among submissions from 24 different groups. We later balanced the recall/precision by including more predictions than just the most confident one in ambiguous cases, and this raised the F-score on the test-set to 42.6%. The new Akane program can be used freely for academic purposes.

1 Introduction

With the increasing number of publications reporting on protein interactions, there is also a steadily increasing interest in extracting information from Biomedical articles by using Natural Language Processing (BioNLP). There has been several *shared tasks* arranged by the BioNLP community to compare different ways of doing such Information Extraction (IE), as reviewed in Krallinger et al.(2008).

Earlier shared tasks have dealt with Protein-Protein Interaction (PPI) in general, but this task focuses on more specific molecular events, such as *Gene_expression, Transcription, Protein_catabolism, Localization and Binding*, plus *(Positive or Negative) Regulation* of proteins or other events. Most of these events are related to PPI, so our hypothesis was that one of the best performing PPI systems would perform well also on this new event extraction task. We decided to modify a publicly available system with flexible configuration scripting (Miwa et al., 2008). Some adjustments had to be made to the existing system, like adding new types of Named Entities (NE) to represent the *events* mentioned above. The modified AkaneRE (for Relation Extraction) can be freely used in academia[1].

2 Material and Methods

The event extraction system is implemented in a pipeline fashion (Fig. 1).

2.1 Tokenization and Sentence Boundary Detection

The text was split into single sentences by a simple sentence detection program, and then each sentence was split into words (tokens). The tokenization was done by using white-space as the token-separator, but since all protein names are known during both training and testing, some extra tokenization rules were applied. For example, the protein

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/∼satre/akane/

Proceedings of the Workshop on BioNLP: Shared Task, pages 103–106,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

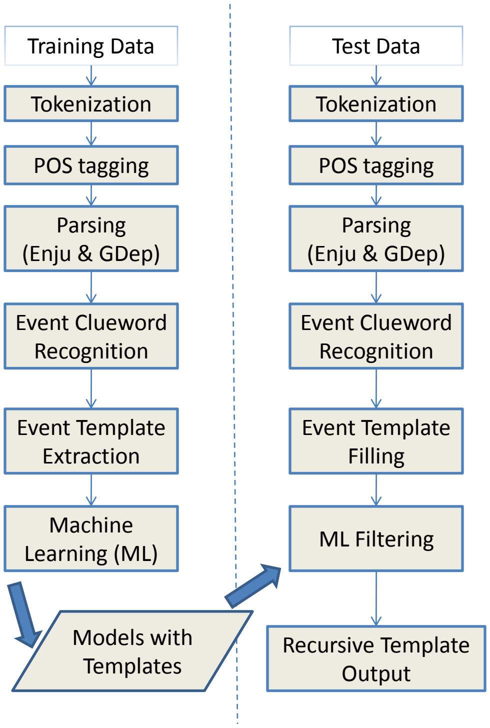

Figure 1: System Overview

name "T cell factor 1" is treated as a single token, "T_cell_factor_1", and composite tokens including a protein name, like "(T_cell_factor_1)", are split into several tokens, like '(', 'T_cell_factor_1' and ')', by adding space around all given protein names. Also, punctuation (*commas*, *periods* etc.) were treated as separate tokens.

2.2 POS-tagging and Parsing

We used Enju[2] and GDep[3] to parse the text. These parsers have their own built-in Part-of-Speech (POS) taggers, and Enju also provides a normalized lemma form for each token.

2.3 Event Clue-word tagging

Event clue-word detection was performed by a Machine Learning (ML) sequence labeling program. This named-entity tagger program is based on a first order Maximum Entropy Markov Model (MEMM) and is described in Yoshida and Tsujii (2007). The clue-word annotation of the shared-task training set was converted into BIO format, and used to train the MEMM model. The features used in the MEMM model was extracted from surface strings and POS information of the words corresponding to (or adjacent to) the target BIO tags. The clue-word tagger was applied to the development and test sets to obtain the marginal probability that each word is a clue-word of a certain category. The probabilities were obtained by marginalizing the n-best output of the MEMM tagger. We later also created clue-word probability annotation of the training set, to enable the template extraction program to access clue-word probability information in the training phase.

2.4 Event Template Extraction

The training data was used to determine which events to extract. As input to the system, a list of *Named Entity* (NE) types and the *Roles* they can play were provided. The roles can be thought of as slots for arguments in event-frames, and in this task the roles were *Event (clue)*, *Theme* and *Cause*. In the original AkanePPI (based on the AIMed corpus), the only NE type was *Protein*, and the only role was *Theme* (p1 and p2). All the (PPI) events were pairwise *interactions*, and there was no explicit *event-clue* role. This means that all the events could be represented with the single template shown first in Table 1.

The BioNLP shared task used eight other NE types, in addition to manually annotated *Proteins*, namely *Binding*, *Gene_expression*, *Localization*, *Protein_catabolism*, *Transcription*, *Regulation*, *Positive_Regulation* and *Negative_Regulation*. The first five events have only *Theme* slots, which can only be filled by *Proteins*, while the last three regulation events are very diverse. They also have one *Theme* slot, but they can have a *Cause* slot as well, and each role/slot can be filled with either *Proteins*, or other *Events*. See the first half of Table 1.

148 templates were extracted and clustered into nine homogeneous groups which were classified as nine separate sub-problems. The grouping was based on whether the templates had an *Event* or a *Protein* in the same role-positions. This way of organizing the groups was motivated by the fact that the *Proteins* are 100% certain, while the accuracy of the clue-word recognizer is only around 50% (estimated on the training data). The bottom of Table 1 shows the resulting nine **general** interaction templates.

[2] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/
[3] http://www.cs.cmu.edu/∼sagae/parser/gdep/

2.5 Machine Learning with Maximum Entropy Models

We integrated Maximum Entropy (ME) modeling, also known as Logistic Regression, into AkaneRE. This was done by using LIBLINEAR[4], which handles multi-class learning and prediction. Gold templates were extracted during training, and each template was matched with all legal combinations of Named Entities (including gold proteins/clue-words and other recognized clue-word candidates) in each sentence. The positive training examples were labeled as gold members of the template, and all other combinations matching a given template were labeled as negative examples within that specific template class. The templates were grouped into the nine *general* templates shown in the bottom of Table 1. Using one-vs-rest logistic regression, we trained one multi-class classifier for each of the nine groups individually. The ML features are shown in Table 2.

In the test-phase, we extracted and labeled all relation candidates matching all the templates from the training-phase. The ML component was automatically run independently for each of the nine groups listed in the bottom of Table 1. Each time, all the candidate template-instances in the current group were assigned a confidence score by the classifier for that group. This score is the probability that a candidate is a true relation, and a value above a certain threshold means that the extracted relation will be predicted as a true member of its specific template. LIBLINEAR's C-value parameter and the prediction threshold were selected by hand to produce a good F-score (according to the strict matching criterion) on the development-test set.

2.6 Filtering and recursive output of the most confident template instances

After machine learning, all the template instances were filtered based on their confidence score. After tuning the threshold to the development test-set, we ended up using 1 as our C-value, and 3.5% as our confidence threshold. Because the prediction of *Regulation Events* were done independent from the sub-events (or proteins) affected by that event, some sub-events had to be included for completeness, even if their confidence score was below the threshold.

3 Results and Discussion

Our final official result was an F-score of 36.9%, which was ranked as number six among the submissions from 24 different groups. This means that the AkanePPI system can achieve good results when used on other PPI-related relation-extraction tasks, such as this first BioNLP event recognition shared task. The most common error was in predicting regulation events with other events as *Theme or Cause*. The problem is that these events involve more than one occurrence of event-trigger words, so the performance is more negatively affected by our imperfect clue-word detection system.

Since the recall was much lower on the test-set than on the development test-set, we later allowed the system to predict multiple confident alternatives for a single event-word, and this raised our score on the test-set from 36.9% to 42.6%. In hindsight, this is obvious since there are many such examples in the training data: E.g. "over-express" is both positive_regulation and Gene_expression. The new system, named AkaneRE (for Relation Extraction), can be used freely for academic purposes.

As future work, we believe a closer integration between the clue-word recognition and the template prediction modules can lead to better performance.

Acknowledgments

"Grant-in-Aid for Specially Promoted Research" and "Genome Network Project", MEXT, Japan.

References

Martin Krallinger et al. 2008. Evaluation of text-mining systems for biology: overview of the second biocreative community challenge. *Genome Biology*, 9(S2).

Makoto Miwa, Rune Sætre, Yusuke Miyao, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Combining multiple layers of syntactic information for protein-protein interaction extraction. In *Proceedings of SMBM 2008*, pages 101–108, Turku, Finland, September.

Kazuhiro Yoshida and Jun'ichi Tsujii. 2007. Reranking for biomedical named-entity recognition. In *Proceedings of the Workshop on BioNLP 2007*, June. Prague, Czech Republic.

[4] http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

Freq	Event	Theme1	Theme2	Theme3	Theme4	Cause
-	PPI	Protein	Protein			
613	Binding	Protein				
213	Binding	Protein	Protein			
3	Binding	Protein	Protein	Protein		
2	Binding	Protein	Protein	Protein	Protein	
217	Regulation	Protein				Protein
12	Regulation	Binding				Protein
48	+Regulation	Transcription				Protein
4	+Regulation	Phosphorylation				Binding
5	-Regulation	+Regulation				Protein
...	**...**	**...**				**...**
Total	148 Templates					
Count	General Templates	**Theme1**	**Theme2**	**Theme3**	**Theme4**	**Cause**
9	event templates	Protein				
1	event template	Protein	Protein			
1	event template	Protein	Protein	Protein		
1	event template	Protein	Protein	Protein	Protein	
3	event templates	Protein				Protein
12	event templates	Protein				Event
27	event templates	Event				
26	event templates	Event				Protein
68	event templates	Event				Event

Table 1: Interaction Templates from the training-set. Classic PPI at the top, compared to Binding and Regulation events in the middle. 148 different templates were automatically extracted from the training data by AkaneRE. At the bottom, the Generalized Interaction Templates are shown, with proteins distinguished from other Named Entities (Events)

Feature	Example
Text	The **binding** of the most prominent factor, named TCF-1 (**T_cell_factor_1**), is correlated with the proto-enhancer activity of TCEd.
BOW_B	The
BOW_M0	-comma- -lparen- factor most named of prominent PROTEIN the
BOW_A	-comma- -rparen- activity correlated is of proto-enhancer the TCEd with
Enju_PATH	(**ENTITY1**) (<prep_arg12arg1) (**of**) (prep_arg12arg2>) (**factor**) (<verb_arg123arg2) (**name**) (verb_arg123arg3>) (**ENTITY2**)
pairs	(**ENTITY1** <prep_arg12arg1) (<prep_arg12arg1 **of**) (**of** prep_arg12arg2>) ...
triples	(**ENTITY1** <prep_arg12arg1 **of**) (<prep_arg12arg1 **of** prep_arg12arg2>) ...
GDep_PATH	(**ENTITY1**) (<NMOD) (**name**) (<VMOD) (**ENTITY2**)
pairs/triples	(**ENTITY1** <NMOD) (<NMOD **name**) ... (**ENTITY1** <NMOD **name**) ...
Vector	BOW_B BOW_M0...BOW_M4 BOW_A Enju_PATH GDep_PATH

Table 2: Bag-Of-Words (BOW) and shortest-path features for the machine learning. Several BOW feature groups were created for each template, based on the position of the words in the sentence, relative to the position of the template's Named Entities (NE). Specifically, BOW_B was made by the words from the beginning of the sentence to the first NE, BOW_A by the words between the last NE and the end of the sentence, and BOW_M0 to BOW_M4 was made by the words between the main event clue-word and the NE in slot 0 through 4 respectively. The path features are made from one, two or three neighbor nodes. We also included certain specific words, like "binding", as features.