Use of clue word annotations as the silver-standard in training models for biological event extraction

Seung-Cheol Baek
Computer Science Department
KAIST
291 Daehak-ro, Yuseong-gu, Daejeon,
305-701, Republic of Korea
scbaek@nlp.kaist.ac.kr

Jong C. Park
Computer Science Department
KAIST
291 Daehak-ro, Yuseong-gu, Daejeon,
305-701, Republic of Korea
park@nlp.kaist.ac.kr

Abstract

Current state-of-the-art approaches to biological event extraction train models by reconstructing relevant graphs from training sentences, where labeled nodes correspond to tokens that indicate the presence of events and the relations between nodes correspond to the relations between these events and their participants. Since multi-word expressions may also indicate events, these approaches use heuristic rules to define target graphs to reconstruct by mapping various clue words into single tokens. Since training instances define actual problems to solve, the method of deriving graphs must affect the system performance, but there has not been any related study on this aspect, to the best of our knowledge. In this study, we propose an incorporation of an EM algorithm into supervised learning to look for training graphs that are more favorable for model construction. We evaluate our algorithm on the development dataset in the 2009 BioNLP shared task and show that this algorithm makes a statistically meaningful improvement on the performance of trained models over a supervised learning algorithm on a fixed set of training graphs.

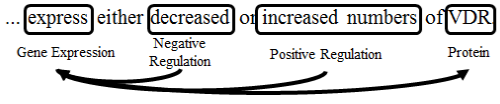
The models and graphs are available at http://biopathway.org/EventExtraction/.

1 Introduction

While the traditional goals of biomedical information extraction include the extraction of named entities and flat binary relations between such entities, there has also been a consistent interest in extracting biological events, which would work as the basis for the subsequent construction of biological pathways (Oda et al., 2008; Ananiadou et al., 2010, Buyko et al., 2011). The 2009 and 2011 BioNLP shared tasks are designed in part in response to these needs (Kim et al., 2009; Kim et al., 2011).

A common approach to biological event extraction is to detect the **clue words** for events and their participants, such as proteins and other events, using statistical models, since event information can be derived from the detected clue words and their relation with simple heuristics (e.g., Björne et al., 2009) and since they are available in various annotated corpora, such as the BioInfer and GENIA Event corpora (Pyysalo et al., 2007; Kim et al., 2008) and the datasets provided by the 2009 and 2011 BioNLP shared tasks (Kim et al., 2009; Kim et al., 2011). Since clue words may be multi-word expressions, the use of dependency paths, which are known to be effective for detecting the participants of events, requires the mapping of clue words into single tokens, or **clue tokens**. The *de facto* standard method, or **head token method**, is to use head tokens within clue words (e.g., Björne et al., 2009). However, our analysis of the dataset in the 2009 BioNLP shared task suggests the presence of favorable tokens other than head tokens.

For example, consider the sentence below (PMID:9783909).

(1) 

The words ‘decreased’ and ‘increased numbers’ are annotated as clue words for two events that either negatively or positively regulate a gene expression event. The head token method maps these words into ‘decreased’ and ‘numbers’, but our intuition is that ‘increased’ is a clue token more favorable than ‘numbers’, since ‘increased’ plays the role of an adjective phrase and conveys the meaning similar to the types of the anchored event, as does the clue token ‘decreased’, and since the shortest dependency path of the token ‘increased’ to ‘express’ is similar to that of the token ‘decreased’.

However, it is unclear not only whether the token ‘increased’ is better than the token ‘numbers’ in terms of system performance, but also whether the use of different clue tokens and their relations affects the system performance in a meaningful way. In this study, we look into their effect on the performance of event extraction models and propose a way to automatically map clue words into clue tokens favorable for models by addressing the 2009 BioNLP shared task 1, which is to identify events and their participants, as a case study.

First, we use the Expectation-Maximization (EM) algorithm with Viterbi approximation, which is often used to address tasks without annotated corpora. Unsurprisingly, our experimental results show that the resulting models show lower performance than our baseline supervised-learning models that use the head token method, because the EM algorithm must adjust models to extract similar but unintended events. To avoid this problem, we modify the EM algorithm, to be called an **informed EM algorithm** in this paper, so that it allows trained models to produce those clue tokens that only encode gold-standard events in the expectation step. The resulting models significantly outperform our baseline supervised-learning models statistically (p-value = 9.59E-12). Thus, we show that different sets of clue tokens and their relations do affect the performance of the respective models.

The rest of the paper is organized as follows. First, we define the task of biological event extraction in Section 2. The subsequent sections (Sections 3, 4, and 5) illustrate our event extraction systems. In Section 6, we present and analyze experimental results. We conclude this paper and present remaining issues in Section 7.

2 Biological Event Extraction

In the present study, we adopt the event extraction task as defined in the 2009 BioNLP shared task 1, which was later renamed as GENIA Event Task 1 and extended to cover full papers in the 2011 BioNLP shared task (Kim et al., 2011). We use biological events to refer to the changes of a state of one or more biological chemicals. Our task is to extract structured information on events from sentences in the biological literature, which consists of their event type and participants encoded with the controlled vocabulary that consists of nine event types.

The nine event types are divided into three groups according to their types of participant, one for events with only single protein arguments (e.g., gene expression and localization events), another for events with multiple protein arguments (e.g., binding events), and the last for events that allow event arguments (e.g., positive and negative regulation events). Extracting the first group of events can be viewed as a binary relation extraction task, while the task of extracting events with a group of participants and nested events distinguishes this task from relation extraction tasks.

3 Event Projection

Inspired by Björne et al. (2009), we address the event extraction task as a graph construction problem for each sentence. In their graph for each sentence, a vertex corresponds to a token and may have a single label indicating that the token is either a clue token for an event or a protein mention. Edges are directed from clue tokens for events to clue tokens for their participants (e.g., other events and proteins) and labeled with the role the participants play (e.g., Theme and Cause). In this study, we use a refined version of their graph representation based on the following analysis of missing events in converting from event annotations in the training dataset into graphs, and vice versa.

While converting event annotations into graphs, we found those events that share clue tokens. Since the representation does not allow vertices and edges to have more than one label, those events cause loss of information. Consider the following sentence (PMID:10202027).

(2) … mRNA and surface expression of E-selectin …

Transcription
Gene expression → Protein

The word ‘expression’ is a clue token for two events: a transcription event, which produces mRNAs, and a gene expression event, which produces proteins. The conjunction ‘and’ combines two virtual phrases, one starting with ‘mRNA expression’, which indicates the transcription event, and the other starting with ‘surface expression’, which indicates the gene expression event. Thus, two events of different types share the word ‘expression’, whose interpretation would be inherently ambiguous.

We also analyzed edges labeled with more than one role, but there were only six edges in the training corpus of 800 annotated abstracts, all of which are apparently annotation noise. Therefore, we modify this graph representation to allow tokens with more than one label, but disallow edges with two or more labels.

In automatically converting graphs into event annotations, we found graphs with cycles that may lead to an infinite number of regulation events with distinct participant events. Consider the sentence below (PMID:10080948), whose corresponding graph has a loop, an extreme case of cycles.

(3) … when transiently overexpressed in 293 cells.

Gene Expression

Positive Regulation

The boxed word is a clue word both for a gene expression event and for a positive regulation event with it as Theme. It would be straightforward to derive these gene expression and regulation events from the graph. The problem is that there is no principled way to rule out another regulation event with the derived regulation event as Theme.

Upon analyzing such loops, we came up with a possible explanation for their presence, which is that the annotators might have failed to find the appropriate type for some events in sentences in the limited controlled vocabulary and would have attempted to use the combination of more than one **component event** to represent the event (**invented events**). In the preceding example, gene expression and regulation events anchored at the word ‘overexpressed’ illustrate examples of invented events. The rest of the loops would be due to tokens hyphenating protein mentions and clue words for events taking the proteins (e.g., ‘IFNgamma-induced’).

Taking into account the fact that invented events have at least one regulation event, which in turn takes another component event as an argument, we found that the combination of gene expression and positive regulation events occurs many times, so we may treat this type of event combinations as single events.

4 Event Extraction Models

In this study, we used a variant of an event extraction model proposed by Riedel and McCallum (2011), which is a joint model for the clue token and relation detection. They proposed three models ranging from the simplest one, Model 1, to the most complicated one, Model 3. Model 3 was ranked second in the GENIA Event subtask of the 2011 BioNLP shared task and its variant (the FAUST system) was ranked first (Riedel et al., 2011). However, we use Model 1 in this study, since Model 3 was reported to be much slower than Model 1 in training and predicting. We trained Model 1 using three different learning algorithms as follows.

As mentioned above, our graph representation is different from theirs in that we allow tokens with more than one label. However, it may not be beneficial to allow models to assign more than one label to single tokens, since there are very few multiply labeled tokens and the number of the possible assignments to single tokens increases exponentially with respect to the number of event types. To understand the effect of multiple labeling, we implemented two distinct types of models, or **multi-labeling** (**ML**) **model** and **single-labeling (SL**) **model**.

4.1 Baseline Models: PA Models

We develop baseline models (**PA models**) by deriving graphs from event annotations using the head token method and training Model 1 on them. In detail, we used an online prediction-based Passive-Aggressive (PA) algorithm (Crammer et al. 2006) with the loss function penalizing false negative labels and edges 3.8 times more than false positive ones (used by Riedel and McCallum (2011)). Inspired by Collins (2002), we constructed models by averaging models’ parameters. As the algorithm takes more passes over the training dataset, the resulting model would perform better on the training dataset with a higher risk of over-fitting the

training dataset. We trained the model for 20 iterations, saving the model after each iteration.

4.2 EM Models

Since the EM approach is widely used in tasks without annotations, we modify this algorithm using the Viterbi approximation approach, or an efficient variant of the EM approach, to produce models (**EM models**) that predict clue tokens other than head tokens. The algorithm starts with a model trained by taking five passes and repeats the process of using the model to predict graphs (the Expectation step) with the graphs for model update (the Maximization step). We implemented this algorithm using a table (**sentence-graph table**) where each row (G_i) corresponds to a sentence and encodes its corresponding graph as sketched in Figure 1. The initial graphs are derived from the event annotations using the head token method. When the predicted graph in the Expectation step is different from that in the table, the algorithm updates the corresponding row. To enforce models to predict clue tokens other than head tokens, we modify the loss function to penalize errors for sentences with updated graphs 10 times more severely than for the others as in domain adaptation works (e.g., Rimell and Clark, 2009). With a PA model trained by taking 5 passes, we trained this model for 15 iterations, saving the model after each iteration.

4.3 Informed EM Models

The EM algorithms may adjust models to predict similar but unintended relations. To reduce this risk, we modify the EM algorithm (**Informed EM algorithm**) so that the algorithm does not allow undesired graphs to enter in the sentence-graph table with the help of constraints. We trained various models (**Informed EM models**) with different sets of constraints. Again, With a PA model trained by taking 5 passes, we trained this model for 15 iterations, saving the model after each iteration.

We used four types of constraints. The first constraint is that graphs in the sentence-graph table encode the same event types and argument types as the graphs derived from the gold-standard event annotations. For example, if a positive regulation event with a gene expression event as Theme appears in the gold-standard annotations, this constraint requires that one or more positive regulation

PROGRAM:
($M_{0,0}$: initial model parameters;
G_i: graphs derived from gold-standard event annotations with the head token method;
D: the training dataset consisting of N sentences.)
FOR t := 1 TO 20:
IF $t \geq 5$: CALL Expectation
CALL Maximization.

SUBROUTINE Expectation:
FOR sentence S_i IN D:
PREDICT G USING M_{t-1}
IF $G \neq G_i$ AND G SATISIFY ALL constraints C_j:
$G_i := G$

SUBROUTINE Maximization:
(Loss: a loss function;
Update: a function producing an updated model)
$M_{t,0} := M_{t,N}$
FOR sentence S_i IN SHUFFLE(D):
PREDICT G USING $M_{t,i-1}$
IF $G \neq G_i$:
IF G_i HAS NOT BEEN UPDATED:
L := Loss(G, G_i)
ELSE:
L := 10 * Loss(G, G_i)
$M_{t,i}$:= Update($M_{t,i-1}$, L, G, G_i)
ELSE: $M_{t,i} := M_{t,i-1}$
M_t := (AVERAGE($M_{t,1} \ldots M_{t,N}$) + (t-1)* M_{t-1}) / t

Figure 1. EM and Informed EM Algorithms (The underline is only for the Informed EM algorithm).

and gene expression events appear in updated graphs and the positive regulation events should take a gene expression event, but does not take care of clue words for these events (e.g., in checking if constructed graphs satisfy this constraint, we represent a positive regulation event anchored at "high levels" in example sentence (4) as "(positive_regulation, Theme: (gene_expression, Theme: c-jun))"). We used this constraint for all Informed EM models, since we believe that event and argument types should be kept.

Another is that the percentage difference in confidence scores between graphs in question and their corresponding graphs in the sentence-graph table should be equal to or greater than a predefined constant α (**confidence constraint constant**).

We came up with two other constraints by analyzing the sentence below (PMID:1313226).

(4) … mRNA, which is … expressed in … at … high levels was … augmented … Gene Expression
Positive Regulation
Positive Regulation

The words 'high levels' and 'augmented' encode two distinct positive regulation events with the same gene expression event as Theme. Since they encode events of the same type with the same participants, those graphs without any one of them do not violate the first two constraints. To avoid such cases, we formulate a conservative constraint (**non-overlapping constraint**, or **NOC** for short) that two distinct clue words (e.g., 'levels' and 'augmented') with the same event type cannot be mapped into a single token (e.g., either 'levels' or 'augmented'). Since this constraint may be too conservative, we came up with a constraint such that the distance between clue words in the constructed graph and clue words labeled with the same event type in the graph (e.g., the distance between 'levels' and 'augmented' is four) should be less than or equal to a predefined constant β (**distance constraint constant**), since these two clue words are distant.

5 Experiments

5.1 Preprocessing

Our system requires lexical information about words and syntactic analyses of sentences. We used analyses of the basic Stanford dependency (SD) generated by the Charniak-Johnson parser (Charniak and Johnson, 2005) with a self-trained biomedical parsing model (McClosky and Charniak, 2008) and the Enju parser with the GENIA model (Miyao et al., 2009), which are available in the official website of BioNLP shared tasks. As for lexical information, we used the base-forms and Part-of-Speech (POS) tags of tokens from the analyses by the Enju parser.

Inspired by Miwa et al. (2010) and Kilicoglu and Bergler (2011), we constructed nine **clue word lexicons**, one for each event type, by collecting tokens from the training dataset. We derived the following entries from tokens within clue words and put them into the lexicons. First, the tokens were put into the lexicon. Second, we split them with hyphens and added the fragments. We also constructed the **stem version** of each clue word lexicon using Porter Stemmer. For entries in these lexicons, we computed the **reliability scores** $G(t, C) = w(C:t)/w(t)$, where $w(C:t)$ is the number of times t occurs within clue words for events of type C and $w(t)$ is the total number of times t occurs, as defined by Krallinger and Berger (2011). We finally removed entries with reliability scores below 1%. After this removal, these lexicons still cover 98% of all clue words in the training dataset, and we only used them to identify the types of their anchored events with precision 10%.

Note that the gold-standard annotations of protein mentions were given to the participants and are available on the official website, so that we used them directly, instead of relying on a named entity recognition system. Some protein mentions are multi-word expressions, so that they require head detection as well. Since we already have protein mentions at our disposal, we use heuristics to replace tokens within protein mentions with single tokens.

5.2 Feature Vectors

For efficiency, we created feature vectors only for those tokens that do not contain any entry in our clue word lexicons and their stem version. We also generated feature vectors only for those edges whose starting token contains any entry in the lexicons and whose ending token is a protein or contains an entry in the lexicons for events that take proteins. Since about 98% of clue words contain an entry in the lexicons, this does not incur a large performance penalty but greatly reduces the size and complexity of the problem.

For clue tokens, we used features for their lexical and linear/syntactic contextual information. Lexical information about tokens is encoded with their surface form, base-form, POS tag and the reliability scores of the entries derived from them in the lexicons. The reliability scores are encoded as both real-valued features and binary features. The linear contextual features that we use include n-grams around them (n = 2 ~ 4) and the position of proteins and the position of potential clue words within the sentences relative to them. The positions of proteins are encoded as binary features, but features for the position of potential clue words take on the maximal reliability score of the corresponding entries in the lexicons. As syntactic contextual features, we encoded their syntactic governors and modifiers (base-form + POS tag).

For pairs of potential clue tokens for events and their arguments, we used a collection of features for the shortest paths between them as taken from Miwa et al. (2010): lengths, n-grams of words, of

pairs of words representing the governor-dependent relationship and of the type and direction of dependency (e.g., >subj), vertex and edge walks, their substructures, syntactical governors and dependents of clue tokens for events and their arguments. In addition, for shortest paths, we use features for the reliability scores of potential clue tokens for events.

5.3 2009 BioNLP Shared Task 1

To measure the consequence of the substitution of single events with the combination of positive regulation and gene expression events sharing single tokens, we converted the gold-standard annotations in the training dataset into graphs and the resulting graphs into event annotations using the heuristics proposed by Björne et al. (2009). We evaluated the resulting event annotations against the gold-standard annotations and found that this substitution increases the F1-score of reconstructed event annotations by 1.13%, and in particular for positive regulation events by 3.14%, though the F1-score for gene expression events is slightly decreased by 0.06%.

We also measure the consequences of allowing a token to have more than one label. We trained our baseline models and evaluated them on the development dataset in terms of standard evaluation metrics, such as recall, precision and F-score (R/P/F).

	PA + SL (R/P/F)	PA + ML (R/P/F)
BEST	46.8/67.0/55.1	47.3/67.7/55.7
AVG. (STD.)	46.2/66.6/54.6 (0.36/0.41/0.32)	46.6/67.1/55.0 (0.23/0.21/0.30)

Table 1. Performance of PA + SL and PA + ML Models.

Table 1 shows the performance of the models of each type. We calculate averages and sample standard deviations using models trained by taking more than five passes to compare with other models that will be discussed later, which are also trained by taking more than five passes. PA + SL models are our implementation of Model 1 of Riedel and McCallum (2011), which was reported to have the F1-score of 56.2, and the best has a similar F1-score of 55.09, where the difference may be due to implementation details regarding the feature-vector construction. As we can see, ML models outperform SL models. Using the one-tailed paired Student's t-test, we find that this superiority of ML models over SL models is statistically significant with a p-value of 0.0013. We also observe that this superiority holds for other types of model. From now on, we present the performance of ML models only for brevity.

Next, we evaluate our EM models. Unsurprisingly, the more passes we took to train models the lower performance the resulting models showed, probably because undesired graphs extensively corrupt the sentence-graph table. As a result, the best one is the model it took six passes to train, which shows a recall of 47.12%, a precision of 67.04% and an F-score of 55.34%. At the first Expectation step, more than a thousand of graphs were updated and at subsequent Expectation steps, fewer than half a hundred graphs were, suggesting that the model is converging (the total number of sentences is about seven thousands), suggesting again that the EM algorithm would have trained models to predict similar but unintended graphs.

We evaluate our Informed EM models as in Table 3, where the best figures are set in boldface.

α=	β=2 (R/P/F)	β=100 (R/P/F)
Without NOC		
0.1	48.0/68.2/56.3	47.6/68.3/56.1
0.2	47.6/68.6/56.2	47.4/68.5/56.0
0.3	47.7/68.8/56.3	47.3/67.5/55.7
0.4	47.1/67.8/55.6	47.6/67.7/55.9
With NOC		
0.1	47.3/68.9/56.1	47.5/68.1/55.9
0.2	47.3/68.0/55.8	47.5/**69.3**/56.4
0.3	**48.1**/68.9/**56.7**	47.2/68.1/55.8
0.4	46.8/68.9/55.8	47.3/67.7/55.7

(a) Best Performance

α=	β=2 (R/P/F)	β=100 (R/P/F)
Without NOC		
0.1	**47.9**/66.8/55.8 (0.27/0.56/0.31)	47.3/67.7/55.7 (0.22/0.30/0.23)
0.2	47.1/68.0/55.7 (0.35/0.86/0.42)	47.1/68.1/55.7 (0.22/0.21/0.16)
0.3	47.4/67.9/55.8 (0.18/0.39/0.23)	47.3/66.8/55.4 (0.13/0.22/0.13)
0.4	46.7/67.5/55.2 (0.38/0.52/0.21)	47.0/67.7/55.5 (0.35/0.23/0.30)
With NOC		
0.1	46.9/68.0/55.5 (0.23/0.39/0.26)	47.1/67.6/55.5 (0.15/0.23/0.16)
0.2	47.1/67.6/55.5 (0.22/0.29/0.20)	47.2/68.3/55.8 (0.22/0.65/0.35)
0.3	47.6/68.0/**56.0** (0.38/0.45/0.40)	47.0/67.1/55.3 (0.27/0.36/0.29)
0.4	46.5/**68.4**/55.4 (0.33/0.72/0.42)	47.1/67.6/55.5 (0.24/0.39/0.22)

(b) Averages and Sample Standard Deviations

Table 2. Performance of Informed EM Models.

Table 2 shows that most of the models outperform the PA models in terms of both the best and averaged F-scores and the other models also outperform the PA models in terms of averaged F-score. To assess the statistical significance of their superiority over PA models in terms of F-score, we use the one-tailed paired Student's t-test to calculate p-values, which show their superiority as shown in Table 3.

α=	β=2 (w.o/w NOC)	β=100 (w.o/w NOC)
0.1	3.32E-09/1.86E-04	1.03E-06/4.47E-06
0.2	9.98E-07/1.21E-08	3.58E-09/1.05E-08
0.3	**9.59E-12**/3.93E-09	4.38E-06/2.95E-03
0.4	4.37E-02/1.19E-04	2.50E-08/6.70E-07

Table 3. p-values for Informed EM Models.

We analyze the effect of constraints. The high confidence constraint constant α reduces the number of updates in the sentence-graph table, making the resulting models similar to PA models as shown in Table 4.

α=	β=2 (w.o/w NOC)	β=100 (w.o/w NOC)
0.1	72/47	98/50
0.2	34/18	46/31
0.3	16/11	25/15
0.4	9/8	9/5

Table 4. Updated Graphs for Informed EM Models.

The distance constraint (β=2) reduces the number of updates in the sentence-graph table and for most times increases the best F-scores but not the averaged F-scores. The NOC also reduces the number of updates but not always increases the best and averaged F-scores. Note that even though our best model is the model we trained with NOC, the best combination of constraints would be with the α value of 0.3 and the β value of 2 and without the NOC as indicated in Table 3.

Upon analyzing updated graphs, we found observed updates of shifting 'clue token' labels from empty words into content words (e.g., 'activity' vs. '-binding' in the noun phrase 'DNA-binding activity' (PMID:9115366)) and from words distant from the participants of the anchored events into words closer to them (e.g., 'simulates' vs. 'activation' in the phrase 'simulates the activation of' (PMID:8557975)). There were also updates of labeling more than one token as clue tokens for a clue word (e.g., 'results' and 'increases' in a phrase starting with 'results in increases of' (PMID:2121746)). Unexpectedly, we found that sets of edges were updated more often than the position of 'clue token' labels. Some edges were copied and redirected (e.g., copies of all edges coming from 'results' are attached to 'increase' in the preceding example) and some edges not used in deriving events from the graphs are removed.

6 Conclusion

In this study, we looked into the effect of clue tokens and relations between them that are different from those derived from gold-standard event annotations with the head token method on the performance of event extraction models by addressing the 2009 BioNLP shared task 1. First, we used an EM algorithm, since the EM algorithm is widely used to address tasks without annotated corpora. Unsurprisingly, our experimental results show that the resulting models have lower performance than our baseline supervised-learning models using the head token method. We then proposed an informed EM algorithm so that it allows trained models to produce those clue tokens that only encode gold-standard events in the Expectation step. The resulting models outperform our baseline models statistically significantly (p-value = 9.59E-12). Thus, we show that there are clue tokens better than head tokens and also propose an automatic way of identifying them.

There are still remaining issues. One is the issue of parameter update scheduling in training. In this study, we fixed the parameters (e.g., α and β) in training. However, Smith and Eisner (2006) show that it would be beneficial for the EM algorithm guided by prior knowledge to soften the constraints, as model parameters are converging. We expect that such update scheduling would also be beneficial for the informed EM algorithm. Second, we applied this approach only to the 2009 BioNLP shared task, but this approach is not dependent on a specific task, so that there is a possibility of applying this approach successfully to tasks in other similar domains, such as Infectious Disease (ID) and Epigenetics and Post-translational Modifications (EPI) domains defined in the 2011 BioNLP shared task, along with quite different domains, such as the newswire domain. We plan to address these issues in the future.

Acknowledgments

We would like to thank the anonymous reviewers for their comments. This work was supported by the National Research Foundation (NRF) of Korea funded by the Ministry of Education, Science and Technology (MEST) (No. 20110029447).

References

S. Ananiadou, S. Pyysalo, J. Tsujii, and D.B. Kell. 2010. Event extraction for systems biology by text mining the literature. *Trends in biotechnology*, 28(7):381–390.

J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 10–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

E. Buyko, J. Linde, S. Priebe, and U. Hahn. 2011. Towards automatic pathway generation from biological full-text publications. *Advances in Intelligent Data Analysis X*, pages 67–79.

E. Charniak and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics.

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

H. Kilicoglu and S. Bergler. 2011. Effective bio-event extraction using trigger words and syntactic dependencies. *Computational Intelligence*, 27(4):583–609.

J.D. Kim, T. Ohta, and J. Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10.

J.D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

J.D. Kim, Y. Wang, T. Takagi, and A. 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.

D. McClosky and E. Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 101–104. Association for Computational Linguistics.

M. Miwa, R. STRE, J.D. Kim, and J. Tsujii. 2010. Event extraction with complex event classification using rich features. *Journal of bioinformatics and computational biology*, 8(1):131.

Y. Miyao, R. Sætre, K. Sagae, T. Matsuzaki, and J. Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. *Proceedings of ACL-08: HLT*, pages 46–54.

K. Oda, J.D. Kim, T. Ohta, D. Okanohara, T. Matsuzaki, Y. Tateisi, and J. Tsujii. 2008. New challenges for text mining: mapping between text and manually curated pathways. *BMC bioinformatics*, 9(Suppl 3):S5.

S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen, and T. Salakoski. 2007. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):50.

S. Riedel and A. McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1–12, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

S. Riedel, D. McClosky, M. Surdeanu, A. McCallum, and C. D. Manning. 2011. Model combination for event extraction in bionlp 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, BioNLP Shared Task '11, pages 51–55, Stroudsburg, PA, USA. Association for Computational Linguistics.

L. Rimell and S. Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of Biomedical Informatics*, 42(5):852–865.

N. A. Smith and J. Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 569–576, Stroudsburg, PA, USA. Association for Computational Linguistics.