

# Syntactic Dependency Based Heuristics for Biological Event Extraction

Halil Kilicoglu and Sabine Bergler

Department of Computer Science and Software Engineering  
Concordia University  
1455 de Maisonneuve Blvd. West  
Montréal, Canada  
{h.kilico, bergler}@cse.concordia.ca

## Abstract

We explore a rule-based methodology for the BioNLP'09 Shared Task on Event Extraction, using dependency parsing as the underlying principle for extracting and characterizing events. We approach the speculation and negation detection task with the same principle. Evaluation results demonstrate the utility of this syntax-based approach and point out some shortcomings that need to be addressed in future work.

## 1 Introduction

Exponential increase in the amount of genomic data necessitates sophisticated approaches to accessing knowledge in molecular biology literature, which remains the primary medium for disseminating new knowledge in molecular biology. Extracting relations and events directly from free text facilitates such access. Advances made in foundational areas, such as parsing and named entity recognition, boosts the interest in biological event extraction (Zweigenbaum et al., 2007). The BioNLP'09 Shared Task on Event Extraction illustrates this shift and is likely to inform future endeavors in the field.

The difficulty of extracting biological events from scientific literature is due to several factors. First, sentences are long and often have long-range dependencies. In addition, the biological processes described are generally complex, involving multiple genes or proteins as well as other biological processes. Furthermore, biological text is rich in higher level phenomena, such as speculation and negation,

which need to be dealt with for correct interpretation of the text. Despite all this complexity, however, a closer look at various biological corpora also suggests that beneath the complexity lie regularities, which may potentially be exploited using relatively simple heuristics.

We participated in Task 1 and Task 3 of the Shared Task on Event Extraction. Our approach draws primarily from dependency parse representation (Mel'čuk, 1988; deMarneffe et al., 2006). This representation, with its ability to reveal long-range dependencies, is suitable for building event extraction systems. Dependencies typed with grammatical relations, in particular, benefit such applications. To detect and characterize biological events (Task 1), we constructed a dictionary of event triggers based on training corpus annotations. Syntactic dependency paths between event triggers and event participants in the training corpus served in developing a grammar for participant identification. For speculation and negation recognition (Task 3), we extended and refined our prior work in speculative language identification, which involved dependency relations as well. Our results show that dependency relations, despite their imperfections, provide a good foundation, on which accurate and reliable event extraction systems can be built and that the regularities of biological text can be adequately exploited with a limited set of syntactic patterns.

## 2 Related Work

Co-occurrence based approaches (Jenssen et al., 2001; Ding et al., 2002) to biological relation extraction provide high recall at the expense of low

precision. Shallow parsing and syntactic templates (Blaschke et al., 1999; Rindflesch et al., 2000; Friedman et al., 2001; Blaschke and Valencia, 2001; Leroy et al., 2003; Ahlers et al., 2007), as well as full parsing (Daraselia et al., 2004; Yakushiji et al., 2005), have also been explored as the basis for relation extraction. In contrast to co-occurrence based methods, these more sophisticated approaches provide higher precision at the expense of lower recall. Approaches combining the strengths of complementary models have also been proposed (Bunescu et al., 2006) for high recall and precision.

More recently, dependency parse representation has found considerable use in relation extraction, particularly in extraction of protein-protein interactions (PPI). Fundel et al. (2007) use Stanford dependency parses of Medline abstracts as the basis for rules that extract gene/protein interactions. Rinaldi et al. (2007) extract relations combining a handwritten grammar based on dependency parsing with a statistical language model. Airola et al. (2008) extract protein-protein interactions from scientific literature using supervised machine learning based on an all-dependency-paths kernel.

The speculative aspect of the biomedical literature (also referred to as *hedging*) has been the focus of several recent studies. These studies primarily dealt with distinguishing speculative sentences from non-speculative ones. Supervised machine learning techniques mostly dominate this area of research (Light et al., 2004; Medlock and Briscoe, 2007; Szarvas, 2008). A more linguistically-based approach, relying on lexical and syntactic patterns, has been explored as well (Kilicoglu and Bergler, 2008). The scope of speculative statements is annotated in the BioScope corpus (Vincze et al., 2008); however, experiments in detecting speculation scope have yet to be reported.

Recognizing whether extracted events are negated is crucial, as negation reverses the meaning of a proposition. Most of the work on negation in the biomedical domain focused on finding negated terms or concepts. Some of these systems are rule-based and rely on lexical or syntactic information (Mutalik et al., 2001; Chapman et al., 2001; Sanchez-Graillet and Poesio, 2007); while others (Averbuch et al., 2004; Goldin and Chapman, 2003) experiment with machine learning techniques. A re-

cent study (Morante et al., 2008) focuses on learning negation scope using memory-based classifiers trained on the BioScope corpus.

Our approach to Task 1 is most similar to work of Fundel et al. (2007) as it builds on dependency-based heuristics. However, we address a larger number of event classes, including regulatory events allowing participation of other events. In addition, event triggers are central to our approach, contrasting with their system and most other PPI systems that rely on finding dependency paths between entities. We extended prior work for Task 3 and obtained state of the art results.

### 3 Event Detection and Characterization

As preparation for biological event extraction, we combined the provided annotations, tokenized input and dependency parses in an XML representation. Next, we determined good trigger words for event classes and scored them. Finally, we developed a dependency-based grammar for event participant identification, which drives our event extraction system.

#### 3.1 Data Preprocessing

Our event detection and characterization pipeline requires XML representation of a document as input. Here, the XML representation of a document contains sentences, their offset positions and dependency parses as well as entities (Proteins) and their offset positions in addition to word information (tokens, part-of-speech tags, indexes and lemmas). We used the Stanford Lexicalized Parser (Klein and Manning, 2003) to extract word-related information, as well as for dependency parsing.

#### 3.2 Event Triggers

After parsing the training corpus and creating an enriched document representation, we proceeded with constructing a dictionary of event triggers, drawing from training corpus annotations of triggers and making further refinements, as described below.

We view event triggers essentially as predicates and thus restricted event triggers to words carrying verb, noun or adjective part-of-speech tags. Our analysis suggests that, in general, trigger words with other POS tags are tenuously annotated event triggers and in fact require more context to qualify as

event triggers. In Example (1), *by* is annotated as trigger for a `Positive_regulation` event; however, it seems that the meaning of the entire prepositional phrase introduced with *by* contributes to trigger such an event:

- (1) These data suggest a general role for Tax induction of IL-1alpha gene transcription *by* the NF-kappaB pathway.

We refined the event trigger list further through limited term expansion and filtering, based on several observations:

1. The event triggers with prefixes, such as *co*, *down* and *up*, (e.g., *coexpression*, *down-regulate*) were expanded to include both hyphenated and non-hyphenated forms.
2. For a trigger that has inflectional/derivational forms acting as triggers in the development corpus but not in the training corpus, we added these forms as event triggers. Examples include adding *dimerization* after *dimerize* and *diminished*(adj) after *diminish*, among others.
3. We removed several event triggers, which, we considered, required more context to qualify as event triggers for the corresponding event classes. (e.g., *absence*, *absent*, *follow*, *lack*)

Finally, we did not consider multi-word event triggers. We observed that core trigger meaning generally came from a single word token (generally head of a noun phrase) in the fragment annotated as event trigger. For instance, for trigger *transcriptional activation*, the annotated event class is `Positive_regulation`, which suggests that the head *activation* carries the meaning in this instance (since *transcriptional* is an event trigger for the distinct `Transcription` event class). In another instance, the trigger *binding activity* is annotated as triggering a `Binding` event, indicating that the head word *activity* is semantically empty. We noted some exceptions to this constraint (e.g., *negatively regulate*, *positive regulation*) and dealt with them in the postprocessing step.

For the remaining event triggers, we computed a “goodness score” via maximum likelihood estimation. For a given event class  $C$  and event trigger  $t$ , the “goodness score”  $G(t,C)$  then is:

$$G(t,C) = w(C:t)/w(t)$$

where  $w(C:t)$  is the number of times  $t$  occurs as a trigger for event class  $C$  and  $w(t)$  is the frequency of trigger  $t$  in the training corpus. The newly added event triggers were assigned the same scores as the trigger they are derived from.

In the event extraction step, we do not consider event triggers with a score below an empirically determined threshold.

### 3.3 Dependency relations for event participant identification

To identify the event participants Theme and Cause, we developed a grammar based on the “collapsed” version of Stanford Parser dependency parses of sentences. Grammar development was driven by extraction and ranking of typed dependency relation paths connecting event triggers to corresponding event participants in the training data. We then analyzed these paths and implemented as rules those deemed to be both correct and sufficiently general.

More than 2,000 dependency paths were extracted; however, their distribution was Zipfian, with approximately 70% of them occurring only once. We concentrated on the most frequent, therefore general, dependency paths. Unsurprisingly, the most frequent dependency path involved the *doj* (direct object) dependency between verbal event triggers and Theme participants, occurring 826 times. Next was the *nn* (nominal modifier) dependency between nominal event triggers and their Theme participants. The most frequent dependency for Cause participants was, again unsurprisingly, *nsubj* (nominal subject). The ranking of dependency paths indicated that path length is inversely proportional to reliability. We implemented a total of 27 dependency path patterns.

Some of these patterns specifically address deficiencies of the Stanford Parser. Prepositional phrases are often attached incorrectly, causing problems in participant identification. Consider, for example, one of the more frequent dependency paths, *doj-prep\_on* (direct object dependency followed by prepositional modifier headed in *on*), occurring between the event trigger (*effect*) and participant (*expression*, itself a sub-event trigger):

- (2) We have examined the *effect* of leukotriene B4

(LTB4), a potent lipid proinflammatory mediator, *on the expression* of the proto-oncogenes *c-jun* and *c-fos*.  
*dobj(examined, effect)*  
*prep\_on(examined, expression)*

This dependency path occurs almost exclusively with PP attachment errors involving *on*, leading us to stipulate a “corrective” dependency path, implemented for certain trigger words (e.g., *effect*, *influence*, *impact* in this case). Postnominal prepositional attachment heuristics detailed in Schuman and Bergler (2006) helped determine 6 such patterns.

Two common verbs (*require* and *involve*) deserve special attention, as the semantic roles of their subject/object constituents differ from typical verbs. The prototypical Cause dependency, *nsubj*, indicates a Theme in the following sentence:

- (3) *Regulation* of interleukin-1beta transcription by Epstein-Barr virus *involves* a number of latent proteins via their interaction with RBP.  
*nsubj(involves, Regulation)*

For these two verbs, participant identification rules are reversed.

An interesting phenomenon is NPs with hyphenated adjectival modifiers, occurring frequently in molecular biology texts (e.g., “... *LPS-mediated TF expression*...”). The majority of these cases involve regulatory events. Such cases do not involve a dependency path, as the participant (in this case, *LPS*) and the event trigger (*mediated*) form a single word. An additional rule addresses these cases, stipulating that the substring preceding the hyphen is the Cause of the regulatory event triggered by the substring following the hyphen. (*Positive\_regulation (Trigger=mediated, Theme=TF expression, Cause=LPS)*).

Events allowing event participants (regulatory events) are treated essentially the same way as events taking entity participants. The main difference is that, when sub-events are considered, a dependency path is found between the trigger of the main event and the trigger of its sub-event, rather than an annotated entity, as was shown above in Example (2).

### 3.4 Extracting Events

The event detection and characterization pipeline (Task 1) consists of three steps:

1. Determining whether a word is an event trigger.
2. If the word is an event trigger, identifying its potential participant(s).
3. If the event trigger corresponds to a regulatory event and it has a potential sub-event participant, determining in a recursive fashion whether the sub-event is a valid event.

The first step is a simple dictionary lookup. Provided that a word is tagged as noun, verb or adjective, we check whether it is in our dictionary, and if so, determine the event class for which it has a score above the given threshold. This word is considered the clue for an event.

We then apply our dependency-based rules to determine whether any entity or event trigger (in the case of regulatory events) in the sentence qualifies as an argument of the event clue. Grammar rules are applied in the order of simplicity; rules that involve a direct dependency between the clue and any word of the entity are considered first.

Once a list of potential participants is obtained by consecutive application of the rules, one of two things may happen: Provided that sub-events are not involved and appropriate participants have been identified (e.g., a Theme is found for a *Localization* event), the event is simply added to the extracted event list. Otherwise, we proceed recursively to determine whether the sub-event participant can be resolved to a simple event. If this yields no such simple event in the end, the event in question is rejected. In the following example, the event triggered by *inhibit* is invalid even though its Cause *JunB* is recognized, because its Theme, sub-event triggered by *activation*, cannot be assigned a Theme and therefore is considered invalid.

- (4) ..., *JunB*, is shown to *inhibit activation* mediated by *JunD*.

After events are extracted in this manner, two postprocessing rules ensure increased accuracy. One rule deals with a limited set of multi-word event triggers. If a *Regulation* event

has been identified and the event trigger is modified by *positive* or *negative* (or inflectional forms *positively*, *negatively*), the event class is updated to `Positive_regulation` or `Negative_regulation`, respectively. The second rule deals with the limitation of not allowing multiple events on the same trigger and adds to the extracted event list a `Positive_regulation` event, if a `Gene_expression` event was recognized for certain triggers, including *overexpression* and several others related to *transfection* (e.g., *transfect*, *transfection*, *cotransfect*).

Two grammatical constructions are crucial to determining the event participants: coordination and apposition. We summarize how they affect event extraction below.

### 3.4.1 Coordination

Coordination plays two important roles in event extraction:

1. When the event trigger is conjoined with another word token, dependency relations concerning the other conjunct are also considered for participant identification.
2. When an event is detected and its participant is found to be coordinated with other entities, new events are created with the event trigger and each of these entities. An exception are `Binding` events, which may have multiple Themes. In this case, we add conjunct entities as the Themes of the base event.

Coordination between words is largely determined by dependency relations. The participants of a dependency with a type descending from *conj* (conjunct) are considered coordinated (e.g., *conj\_and*, *conj\_or*).

Recognizing that Stanford dependency parsing misses some expressions of coordinated entities typical of biological text (in particular, those involving parentheses), we implemented a few additional rules to better resolve coordinated entities. These rules stipulate that entities that have between them:

1. Only a comma (,) or a semi-colon (;)
2. A word with CC (coordinating conjunction) part-of-speech tag

3. A complete parenthetical expression

4. Any combination of the above

are coordinated. For instance, in Example (5), we recognize the coordination between *interleukin-2* and *IL-4*, even though the parser does not:

- (5) The activation of NFAT by TCR signals has been well described for *interleukin-2* (IL-2) and *IL-4* gene transcription in T cells.  
*conj\_and(interleukin-2,transcription)*

### 3.4.2 Apposition

Words in an apposition construction are considered equivalent for event extraction purposes. Therefore, if an appropriate dependency exists between a word and the trigger and the word is in apposition with an entity, that entity is marked as the event participant. In Example 6, the *appos* (appositive) dependency shown serves to extract the event `Positive_regulation` (Trigger=*upregulation*, Theme=*intercellular adhesion molecule-1*):

- (6) ... *upregulation* of the lung vascular adhesion molecule, *intercellular adhesion molecule-1*, was greatly reduced by...  
*appos(molecule,molecule-1)*  
*prep\_of(upregulation,molecule)*

The dependencies that we consider to encode apposition constructions are: *appos* (appositive), *abbrev* (abbreviation), *prep\_{including, such\_as, compared\_to, compared\_with, versus}* (prepositional modifier marked with *including*, *such as*, *compared to*, *compared with* or *versus*).

## 3.5 Speculation and Negation Detection

Once an event list is obtained for a sentence, our speculation and negation module determines whether these events are speculated and/or negated, using additional dependency-based heuristics that consider the dependencies between the event trigger and speculation/negation cues.

### 3.5.1 Speculation Recognition

We refined an existing speculation detection module in two ways for Task 3. First, we noted that modal verbs (e.g., *may*) and epistemic adverbs (e.g., *probably*) rarely mark speculative contexts in the

training corpus, demonstrating the lack of a standardized notion of speculation among various corpora. For Task 3, we ignored lexical cues in these classes completely for increased accuracy. Secondly, corpus analysis revealed a new syntactic pattern for speculation recognition. This pattern involves the class of verbs that we called *active cognition verbs* (e.g., *examine*, *evaluate*, *analyze*, *study*, *investigate*). We search for a Theme dependency pattern between one of these verbs and an event trigger and mark the event as speculated, if such a pattern exists. Nominalizations of these verbs are also considered. In Example (7), the event triggered by *effects* is speculated, since *effects* is the direct object (therefore, Theme) of *studied*:

- (7) We have *studied* the *effects* of prednisone (PDN), ... on the production of cytokines (IL-2, IL-6, TNF-alpha, IL-10) by peripheral T lymphocytes...

### 3.5.2 Negation Detection

Negation detection is similar to speculation detection. Several classes of negation cues have been determined based on corpus analysis and the negation module negates events if there is an appropriate dependency between one of these cues and the event triggers. The lexical cues and the dependencies that are sought are given in Table 1.

Negation Cue	Dependency
<i>lack</i> , <i>absence</i>	<i>prep_of(Cue, Trigger)</i>
<i>unable</i> , <i>&lt;not&gt;</i> <i>able</i> , <i>fail</i>	<i>xcomp(Cue, Trigger)</i>
<i>inability</i> , <i>failure</i>	<i>infmod(Cue, Trigger)</i>
<i>no</i> , <i>not</i> , <i>cannot</i>	<i>det(Trigger, Cue)</i>

Table 1: Negation cues and the corresponding dependencies (*xcomp*: clausal complement, *infmod*: infinitival modifier, *det*: determiner)

Additionally, participation of event triggers in dependencies of certain types is sufficient for negating the event it triggers. Such dependency types are *neg* (negation) and *conj\_negcc* (negated coordination). A *neg* dependency applies to event triggers only, while *conj\_negcc* is sought between event participants, as well as event triggers. Therefore, in Example (8), an event (*Positive\_regulation(Trigger=transactivate,*

Theme: *GM-CSF*, Cause=*ELF1*)) is negated, based on the dependencies below:

- (8) Exogenous *ETS1*, but not *ELF1*, can *transactivate GM-CSF*, ..., in a PMA/ionomycin dependent manner.  
*conj\_negcc(ETS1, ELF1)*  
*nsubj(transactivate, ETS1)*  
*doobj(transactivate, GM-CSF)*

Finally, if none of the above applies and the word preceding the event trigger or one of the event participants is a negation cue (*no*, *not*, *cannot*), the event is negated.

## 4 Results and Discussion

Our event extraction system had one of the best performances in the shared task. With the approximate span matching/approximate recursive matching evaluation criteria, in Task 1, we were ranked third, while our speculation and negation detection module performed best among the six participating systems in Task 3. Not surprisingly, our system favors precision, typical of rule-based systems. Full results are given in Table 2.

The results reported are at goodness score threshold of .08. Increasing the threshold increases precision, while lowering recall. The threshold was determined empirically.

Our results confirm the usefulness of dependency relations as foundation for event extraction systems. There is much room for improvement, particularly in terms of recall, and we believe that incremental nature of our system development accommodates such improvements fairly easily.

Our view of event triggers (“once a trigger, always a trigger”), while simplistic, provides a good starting point by greatly reducing the number of trigger candidates in a sentence and typed dependencies to consider. However, it also leads to errors. One such example is given in Example (9):

- (9) We show that ..., and that LPS treatment enhances the *oligomerization* of *TLR2*.

where we identify the event Binding (Trigger=*oligomerization*, Theme=*TLR2*). We consider *oligomerization* a reliable trigger, since it occurs twice in the training corpus, both times as event triggers. However, in this instance, it does not trigger

Event Class	Recall	Precis.	F-score
Localization	35.63	92.54	51.45
Binding	20.46	40.57	27.20
Gene_expression	55.68	79.45	65.47
Transcription	15.33	60.00	24.42
Protein_catabolism	64.29	56.25	60.00
Phosphorylation	69.63	95.92	80.69
EVT-TOTAL	43.10	73.47	54.33
Regulation	24.05	45.75	31.53
Positive_regulation	28.79	50.45	36.66
Negative_regulation	26.65	51.53	35.13
REG-TOTAL	27.47	49.89	35.43
Negation	14.98	50.75	23.13
Speculation	16.83	50.72	25.27
MOD-TOTAL	15.86	50.74	24.17
ALL-TOTAL	32.68	60.83	42.52

Table 2: Evaluation results

an event. This narrow view also leads to recall errors, in which we do not recognize an event trigger as such, simply because we have not encountered it in the training corpus, or it does not have an appropriate part-of-speech tag. A more sophisticated trigger learning approach could aid in better detecting event triggers.

We dealt with some deficiencies of Stanford dependency parsing through additional rules, as described in Section 3.3. However, many dependency errors are still generated, due to the complexity of biological text. For instance, in Example (10), there is a coordination construction between *NF-kappaB nuclear translocation* and *transcription of E-selectin and IL-8*. However, this construction is missed and an erroneous *prep\_of* dependency is found, leading to two false positive errors: *Localization* (Trigger=*translocation*, Theme=*E-selectin*) and *Localization* (Trigger=*translocation*, Theme=*IL-8*).

- (10) ... leading to *NF-kappaB nuclear translocation* and *transcription of E-selectin and IL-8*, which results in ...  
*conj\_and(transcription, translocation)*  
*prep\_of(translocation, E-selectin)*

*conj\_and(E-selectin, IL-8)*

These errors can be corrected via other “corrective” dependency paths; however, first, a closer examination of such error patterns is necessary.

In other instances, the required dependency is completely missed by the parser, leading to recall errors. For instance, in Example (11), we are unable to recognize two events (*Regulation* (Trigger=*regulation*, Theme=*4E-BP1*) and *Regulation* (Trigger=*regulation*, Theme=*4E-BP2*)), due to lack of apposition dependencies between *repressors* and *4E-BP1* or *4E-BP2*:

- (11) ... specific *regulation* of two repressors of translation initiation, *4E-BP1* and *4E-BP2*.  
*prep\_of(regulation, repressors)*  
*prep\_of(repressors, initiation)*  
*conj\_and(intiation, 4E-BP1)*  
*conj\_and(intiation, 4E-BP2)*

Typical of rule-based systems, we miss events expressed using less frequent patterns. Event participants expressed as prepositional modifiers marked with *from* is one such case. An example is given below:

- (12) Calcineurin activates *transcription* from the *GM-CSF* promoter ...

In this case, the event *Transcription* (Trigger=*transcription*, Theme=*GM-CSF*) is missed. It is fairly easy to add a rule to address such occurrences.

We have not attempted to resolve anaphoric expressions for the shared task, which led to a fair number of recall errors. In a similar vein, we ignored events spanning multiple sentences. We expect that several studies addressing anaphora resolution in biomedical text (Castaño et al., 2002; Gasperin and Briscoe, 2008) will inform our near future efforts in this area.

Evaluation results regarding Task 3 may seem poor at first; however, most of the errors concern misidentified or missed base events. Thus, in this section, we focus on errors specifically triggered by speculation and negation module. In the development corpus, we identified 39 speculation instances, 4 of which were errors due to speculation processing. Of 95 annotated speculation instances, 7 were missed due to deficiencies in speculation processing.

Similarly, negation processing led to 5 false positives in 31 negation instances we identified and to 5 false negatives in 107 annotated negation instances.

We found that speculation false positive errors are exclusively cases for which speculation could be argued. For instance, in Example (13), we recognize that *appears to* scopes over event `Negative_regulation` (Trigger=*negatively regulate*, Theme=*IL-2R*), rendering it speculative. However, it is not annotated as such. This is further evidence for the difficulty of annotating such phenomena correctly and consistently, since the exact meaning is somewhat elusive.

- (13) An unidentified Ets family protein binds to the EBS overlapping the consensus GAS motif and *appears to negatively regulate* the human IL-2R alpha promoter.

Negation pattern that involves negation cues (*no, not, cannot*) in the token preceding an event trigger or participant, a pattern initially considered to increase recall, caused most of negation false positive errors. An example is given in (14):

- (14) The finding that HL-60/vinc/R cells respond to TPA with induction of a monocytic phenotype, but *not c-jun expression*, suggests that ...

Complex and less frequent patterns of expressing speculation and negation were responsible for more recall errors. Two such examples are given below:

- (15) (a) These results ... and *suggest* a molecular mechanism for the *inhibition* of TLR2 by DN variants.  
(b) *Galectin-3* is ... and is *expressed* in many leukocytes, *with the notable exception* of B and T lymphocytes.

In (15a), speculation is detected; however, we are unable to recognize that it scopes over the event triggered by *inhibition*. In (15b), the prepositional phrase, *with the notable exception*, is not considered to indicate negation.

## 5 Conclusions and Future Work

We explored a rule-based approach to biological event detection driven by typed dependency relations. This study marks our first foray into bio-event

extraction in a general way and, thus, we consider the results very encouraging. In one area we investigated before, speculation detection, our system performed best and this confirms the portability and extensibility of our approach.

Modest recall figures point to areas of improvement. We plan to address anaphora resolution and multiple sentence spanning events in the near future. Our naïve approach to event triggers needs refinement and we believe that sophisticated supervised machine learning techniques may be helpful. In addition, biomedical lexical resources, including UMLS SPECIALIST Lexicon (McCray et al., 1994), may be useful in improving event trigger detection. Finally, dependency relations based on the Stanford Parser provided better performance in our case, in contrast to general consensus that those based on Charniak Parser (Charniak and Johnson, 2005) are superior, and this, too, deserves further investigation.

## References

- C B Ahlers, M Fiszman, D Demner-Fushman, F M Lang, and T C Rindfleisch. 2007. Extracting semantic predications from Medline citations for pharmacogenomics. *Pac Symp Biocomput*, pages 209–220.
- A Airola, S Pyysalo, J Björne, T Pahikkala, F Ginter, and T Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, 9 Suppl 11:s2.
- M Averbuch, T Karson, B Ben-Ami, O Maimon, and L Rokach. 2004. Context-sensitive medical information retrieval. In *Proc MEDINFO-2004*, pages 1–8.
- C Blaschke and A Valencia. 2001. The potential use of SUISEKI as a protein interaction discovery tool. *Genome Inform*, 12:123–134.
- C Blaschke, M A Andrade, C Ouzounis, and A Valencia. 1999. Automatic extraction of biological information from scientific text: protein-protein interactions. In *Proc Int Conf Intell Syst Mol Biol*, pages 60–67.
- R Bunescu, R Mooney, A Ramani, and E Marcotte. 2006. Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from Medline. In *Proc BioNLP Workshop on Linking Natural Language Processing and Biology*, pages 49–56.
- J Castaño, J Zhang, and J Pustejovsky. 2002. Anaphora resolution in biomedical literature. In *Proc International Symposium on Reference Resolution for NLP*.



- W W Chapman, W Bridewell, P Hanbury, G F Cooper, and B G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform*, 34(5):301–310.
- E Charniak and M Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proc 43rd Meeting of the Association for Computational Linguistics*, pages 173–180.
- N Daraselia, A Yuryev, S Egorov, S Novichkova, A Nikitin, and I Mazo. 2004. Extracting human protein interactions from MEDLINE using a full-sentence parser. *Bioinformatics*, 20(5):604–611.
- M C deMarneffe, B MacCartney, and C D Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc 5th International Conference on Language Resources and Evaluation*, pages 449–454.
- J Ding, D Berleant, D Nettleton, and E Wurtele. 2002. Mining MEDLINE: abstracts, sentences, or phrases? *Pac Symp Biocomput*, 7:326–337.
- C Friedman, P Kra, M Krauthammer, H Yu, and A Rzhetsky. 2001. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17(1):74–82.
- K Fundel, R Küffner, and R Zimmer. 2007. RelEx relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.
- C Gasperin and T Briscoe. 2008. Statistical anaphora resolution in biomedical texts. In *Proc COLING 2008*.
- I M Goldin and W W Chapman. 2003. Learning to detect negation with not in medical texts. In *Proc Workshop on Text Analysis and Search for Bioinformatics at the 26th ACM SIGIR Conference*.
- T K Jenssen, A Laegreid, J Komorowski, and E Hovig. 2001. A literature network of human genes for high-throughput analysis of gene expression. *Nat Genet*, 28:21–28.
- H Kilicoglu and S Bergler. 2008. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinformatics*, 9 Suppl 11:s10.
- D Klein and C D Manning. 2003. Accurate unlexicalized parsing. In *Proc 41th Meeting of the Association for Computational Linguistics*, pages 423–430.
- G Leroy, H Chen, and J D Martinez. 2003. A shallow parser based on closed-class words to capture relations in biomedical text. *Journal of Biomedical Informatics*, 36:145–158.
- M Light, X Y Qiu, and P Srinivasan. 2004. The language of bioscience: facts, speculations, and statements in between. In *BioLINK 2004: Linking Biological Literature, Ontologies and Databases*, pages 17–24.
- A T McCray, S Srinivasan, and A C Browne. 1994. Lexical methods for managing variation in biomedical terminologies. In *Proc 18th Annual Symposium on Computer Applications in Medical Care*, pages 235–239.
- B Medlock and T Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proc 45th Meeting of the Association for Computational Linguistics*, pages 992–999.
- I A Mel’čuk. 1988. *Dependency syntax: Theory and Practice*. State University Press of New York, NY.
- R Morante, A Liekens, and W Daelemans. 2008. Learning the scope of negation in biomedical text. In *Proc Conference on Empirical Methods in Natural Language Processing*, pages 715–724.
- P G Mutalik, A Deshpande, and P M Nadkarni. 2001. Use of general-purpose negation detection to augment concept indexing of medical documents: A quantitative study using the UMLS. *J Am Med Inform Assoc*, 8(6):598–609.
- F Rinaldi, G Schneider, K Kaljurand, M Hess, C Andronis, O Konstandi, and A Persidis. 2007. Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach. *Artif. Intell. Med.*, 39(2):127–136.
- T C Rindflesch, L Tanabe, J N Weinstein, and L Hunter. 2000. EDGAR: Extraction of drugs, genes, and relations from the biomedical literature. In *Proc Pacific Symposium on Biocomputing*, pages 514–525.
- O Sanchez-Graillet and M Poesio. 2007. Negation of protein protein interactions: analysis and extraction. *Bioinformatics*, 23(13):424–432.
- J Schuman and S Bergler. 2006. Postnominal prepositional phrase attachment in proteomics. In *Proc BioNLP Workshop on Linking Natural Language Processing and Biology*, pages 82–89.
- G Szarvas. 2008. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proc 46th Meeting of the Association for Computational Linguistics*, pages 281–289.
- V Vincze, G Szarvas, R Farkas, G Mora, and J Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9 Suppl 11:S9.
- A Yakushiji, Y Miyao, Y Tateisi, and J Tsujii. 2005. Biomedical event extraction with predicate-argument structure patterns. In *Proc First International Symposium on Semantic Mining in Biomedicine*, pages 60–69.
- P Zweigenbaum, D Demner-Fushman, H Yu, and K B Cohen. 2007. Frontiers of biological text mining: current progress. *Briefings in Bioinformatics*, 8(5):358–375.