

# Using SVMs with the Command Relation Features to Identify Negated Events in Biomedical Literature

**Farzaneh Sarafraz**

School of Computer Science  
University of Manchester  
Manchester, United Kingdom  
sarafraf@cs.man.ac.uk

**Goran Nenadic**

School of Computer Science  
University of Manchester  
Manchester, United Kingdom  
g.nenadic@manchester.ac.uk

## Abstract

In this paper we explore the identification of negated molecular events (e.g. protein binding, gene expressions, regulation, etc.) in biomedical research abstracts. We construe the problem as a classification task and apply a machine learning (ML) approach that uses lexical, syntactic, and semantic features associated with sentences that represent events. Lexical features include negation cues, whereas syntactic features are engineered from constituency parse trees and the *command* relation between constituents. Semantic features include event type and participants. We also consider a rule-based approach that uses only the command relation. On a test dataset, the ML approach showed significantly better results (51% F-measure) compared to the command-based rules (35-42% F-measure). Training a separate classifier for each event class proved to be useful, as the micro-averaged F-score improved to 63% (with 88% precision), demonstrating the potential of task-specific ML approaches to negation detection.

## 1 Introduction

With almost 2000 new papers published every day, biomedical knowledge is mainly communicated through a growing body of research papers. As the amount of textual information increases, the need for sophisticated information extraction (IE) methods are becoming more than evident. IE methods rely on a range of language processing methods such as named entity recognition and parsing to extract the required information in a more structured form which can be used for knowledge exploration and hypothesis generation (Donaldson et al. 2003; Natarajan et al. 2006).

Given the large number of publications, the identification of conflicting or contradicting facts

is critical for systematic mining of biomedical literature and knowledge consolidation. Detection of negations is of particular importance for IE methods, as it often can hugely affect the quality of the extracted information. For example, when mining molecular events, a key piece of information is whether the text states that the two proteins *are* or *are not* interacting, or that a given gene *is* or *is not* expressed. In recent years, several challenges and shared tasks have included the extraction of negations, typically as part of other tasks (e.g. the BioNLP'09 Shared Task 3 (Kim et al. 2009)).

Several systems and methods have aimed to handle negation detection in order to improve the quality of extracted information (Hakenberg et al. 2009; Morante and Daelemans 2009). Prior research on this topic has primarily focused on finding negated concepts by negation cues and scopes. These concepts are usually represented by a set of predefined terms, and negation detection typically aims to determine whether a term falls within the scope of a negation cue.

In this paper we address the task of identification of negated events. We present a machine learning (ML) method that combines a set of features mainly engineered from a sentence parse tree with lexical cues. More specifically, parse-based features use the notion of the *command* relation that models the scope affected by an element (Langacker, 1969). We use molecular events as a case study and experiment on the BioNLP'09 data, which comprises a gold-standard corpus of research abstracts manually annotated for events and negations (Kim et al. 2009). The evaluation shows that, by using the proposed approach, negated events can be identified with precision of 88% and recall of 49% (63% F-measure). We compare these results with two rule-based approaches that achieved the maximum F-measure of 42%.

The rest of this paper is organised as follows. Section 2 summarises and reviews previous research on negation extraction. Section 3 defines the problem and introduces the data used for the case study. Section 4 focuses on the ML-based methodology for extracting negated events. The final sections contain the results and discussions.

## 2 Related Work

There have been numerous contemplations of the concept of negation (Lawler, 2010), but no general agreement so far exists on its definition, form, and function. We adopt here a definition of negation as given by Cambridge Encyclopedia of Language Sciences: “Negation is a comparison between a ‘real’ situation lacking some element and an ‘imaginal’ situation that does not lack it”. The imaginal situation is **affirmative** compared with the **negative** real situation. The element whose polarity differs between the two situations is the negation **target**.

Negations in natural language can be expressed by syntactically negative expressions, i.e. with the use of negating words such as *no*, *not*, *never*, etc. The word or phrase that makes the sentence wholly or partially negative is the negation **cue** and the part of the sentence that is affected by the negation cue and has become negative is the negation **scope**.

We briefly review two classes of approaches to detect negations: those aiming at negated concepts and those targeting negated events.

### 2.1 Detecting Negated Concepts and Phrases

There have been a number of approaches suggested for detection of negated targets and scopes. Most of them rely on task-specific, hand-crafted rules of various complexities. They differ in the size and composition of the list of negation cues, and the way to utilise such a list. Some methods use parse trees, whilst others use results of shallow parsing.

Rule-based methods range from simple co-occurrence based approaches to patterns that rely on shallow parsing. The ‘bag-of-words’ approach, looking for proximate co-occurrences of negation cues and terms in the same sentence, is probably the simplest method for finding negations, and is used by many as a baseline method.

Many approaches have targeted the clinical and biomedical domains. NegEx (Chapman et al. 2001), for example, uses two generic regular ex-

pressions that are triggered by negation phrases such as:

$$\begin{aligned} &\langle \text{negation cue} \rangle * \langle \text{target term} \rangle \\ &\langle \text{target term} \rangle * \langle \text{negation cue} \rangle \end{aligned}$$

where the asterisk (\*) represents a string of up to five tokens. Target terms represent domain concepts that are terms from the Unified Medical Language System (UMLS<sup>1</sup>). The cue set comprises 272 clinically-specific negation cues, including those such as *denial of* or *absence of*. Although simple, the proposed approach showed good results on clinical data (78% sensitivity (recall), 84% precision, and 94% specificity).

In addition to concepts that are explicitly negated by negation phrases, Patrick et al. (2006) further consider so-called pre-coordinated negative terms (e.g. *headache*) that have been collected from SNOMED CT<sup>2</sup> medical terminology. Similarly, NegFinder uses hand-crafted rules to detect negated UMLS terms, including simple conjunctive and disjunctive statements (Mutalik et al. 2001). They used a list of 60 negation cues. Tolentino et al. (2006), however, show that using rules on a small set of only five negation cues (*no*, *neither/nor*, *ruled out*, *denies*, *without*) can still be reasonably successful in detecting negations in medical reports (F-score 91%).

Huang and Lowe (2007) introduced a negation grammar that used regular expressions and dependency parse trees to identify negation cues and their scope in the sentence. They applied the rules to a set of radiology reports and reported a precision of 99% and a recall of 92%.

Not many efforts have been reported on using machine learning to detect patterns in sentences that contain negative expressions. Still, Morante and Daelemans (2009), for example, used various classifiers (Memory-based Learners, Support Vector Machines, and Conditional Random Fields) to detect negation cues and their scope. An extensive list of features included the token’s stem and part-of-speech, as well as those of the neighbouring tokens. Separate classifiers were used for detecting negation cues and negation scopes. The method was applied to clinical text, biomedical abstracts, and biomedical papers with F-scores of 80%, 77%, and 68% respectively.

### 2.2 Detecting Negated Events

Several approaches have recently been suggested for the extraction of negated events, particularly

<sup>1</sup> <http://www.nlm.nih.gov/research/umls/>

<sup>2</sup> <http://www.snomed.org>

in the biomedical domain. Events are typically represented via *participants* (biomedical entities that take part in an event) and event *triggers* (tokens that indicate presence of the event). Van Landeghem et al. (2008) used a rule-based approach based on token distances in sentence and lexical information in event triggers to detect negated molecular events. Kilicoglu and Bergler (2009), Hakenberg et al. (2009), and Sanchez (2007) used a number of heuristic rules concerning the type of the negation cue and the type of the dependency relation to detect negated molecular events described in text. For example, a rule can state that if the negation cue is “lack” or “absence”, then the trigger has to be in the prepositional phrase of the cue; or that if the cue is “unable” or “fail”, then the trigger has to be in the clausal complement of the cue (Kilicoglu and Bergler 2009). As expected, such approaches suffer from lower recall.

MacKinlay et al. (2009), on the other hand, use ML, assigning a vector of complex deep parse features (including syntactic predicates to capture negation scopes, conjunctions and semantically negated verbs) to every event trigger. The system achieved an F-score of 36% on the same dataset as used in this paper.

We note that the methods mentioned above mainly focus on finding negated triggers in order to detect negated events. In this paper we explore not only negation of triggers but also phrases in which participants are negated (consider, for example, “SLP-76” in the sentence “*In contrast, Grb2 can be coimmunoprecipitated with Sos1 and Sos2 but not with SLP-76.*”)

### 3 Molecular Events

As a case study, we look at identification of negated molecular events. In general, molecular events include various types of reactions that affect genes and protein molecules. Each event is of a particular *type* (e.g. binding, phosphorylation, regulation, etc.). Depending on the type,

each event may have one or more participating proteins (sometimes referred to as *themes*). Regulatory events are particularly complex, as they can have a *cause* (a protein or another event) in addition to a theme, which can be either a protein or another event. Table 1 shows examples of five events, where participants are biomedical entities (events 1-3) or other events (events 4 and 5). Note that a sentence can express more than one molecular event.

Identification of molecular events in the literature is a challenging IE task (Kim et al. 2009; Sarafranz et al. 2009). For the task of identifying negated events, we assume that events have already been identified in text. Each event is represented by its type, a textual trigger, and one or more participants or causes (see Table 1). Since the participants of different event types can vary in both their number and type, we consider three classes of events to support our analysis (see Section 5):

- Class I comprises events with exactly one entity theme (e.g. transcription, protein catabolism, localization, gene expression, phosphorylation).
- Class II events include binding events only, which have one or more entity participants.
- Class III contains regulation events, which have exactly one theme and possibly one cause. However, the theme and the cause can be entities or events of any type.

The corpus used in this study is provided by the BioNLP’09 challenge (Kim et al. 2009). It contains two sets of biomedical abstracts: a “training” set (containing 800 abstracts used for training and analysis purposes) and a “development” set (containing 150 abstracts used for testing purposes only). Both document sets are manually annotated with information about entity mentions (e.g. genes and proteins). Sentences that report molecular events are further annotated with the corresponding event type, textual trigger and participants. In total, nine event types are

“The effect of this synergism was perceptible at the level of induction of the IL-2 gene.”				
Event	Trigger	Type	Participant (theme)	Cause
Event 1	“induction”	Gene expression	IL-2	

  

“Overexpression of full-length ALG-4 induced transcription of FasL and, consequently, apoptosis.”				
Event	Trigger	Type	Participant (theme)	Cause
Event 2	“transcription”	Transcription	FasL	
Event 3	“overexpression”	Gene expression	ALG-4	
Event 4	“Overexpression”	Positive regulation	Event 3	
Event 5	“induced”	Positive regulation	Event 2	Event 4

Table 1: Examples of how molecular events described in text are characterised.

considered (gene expression, transcription, protein catabolism, localization, phosphorylation, binding, regulation, positive regulation, and negative regulation). In addition, every event has been tagged as either affirmative (reporting a specific interaction) or negative (reporting that a specific interaction has not been observed).

Table 2 provides an overview of the two BioNLP’09 datasets. We note that only around 6% of events are negated.

Event class	Training data		Development data	
	total	negated	total	negated
Class I	2,858	131	559	26
Class II	887	44	249	15
Class III	4,870	440	987	66
<b>Total</b>	<b>9,685</b>	<b>615</b>	<b>1,795</b>	<b>107</b>

Table 2: Overview of the total number of events and negated event annotations in the two datasets.

## 4 Methodology

We consider two approaches to extract negated events. We first discuss a rule-based approach that uses constituency parse trees and the *command* relation to identify negated events. Then, we introduce a ML method that combines lexical, syntactic and semantic features to identify negated events. Note that in all cases, input sentences have been pre-annotated for entity mentions, event triggers, types, and participants.

### 4.1 Negation Detection Using the *Command* Relation Rules

The question of which parts of a syntactic structure affect the other parts has been extensively investigated. Langacker (1969) introduced the concept of *command* to determine the scope within a sentence affected by an element. More precisely, if  $a$  and  $b$  are nodes in the constituency parse tree of a sentence, then  $a$  X-commands  $b$  iff the lowest ancestor of  $a$  with label X is also an ancestor of  $b$ . Note that the command relation is not symmetrical. Langacker observed that when  $a$  S-commands  $b$ , then  $a$  affects the scope containing  $b$ . For simplicity, we say “command” when we mean S-command.

To determine whether token  $a$  commands token  $b$ , given the parse tree of a sentence, we use a simple algorithm introduced by McCawley (1993): trace up the branches of the constituency parse tree from  $a$  until you hit a node that is labelled X. If  $b$  is reachable by tracing down the

branches of the tree from that node, then  $a$  X-commands  $b$ ; otherwise, it does not.

We hypothesise that if a negation cue commands an event trigger *or* participant, then the associated event is negated.

### 4.2 Negation Detection Using Machine Learning on Parse Tree Features

Given a sentence that describes an event, we further construe the negation detection problem as a classification task: the aim is to classify the event as affirmative or negative. We explore both a single SVM (support vector machine) classifier for all events and three separate SVMs for each of the event classes. The following features have been engineered from an event-representing sentence:

1. Event type (one of the nine types as defined in BioNLP’09);
2. Whether the sentence contains a negation cue from the cue list;
3. The negation cue itself (if present);
4. The part-of-speech (POS) tag of the negation cue;
5. The POS tag of the event trigger;
6. The POS tag of the participants of the event. If the participant is another event, the POS tag of the trigger of that event is used;
7. The parse node type of the lowest common ancestor of the trigger and the cue (i.e. the type of the smallest phrase that contains both the trigger and the cue, e.g. S, VP, PP, etc.);
8. Whether or not the negation cue commands any of the participants; nested events (for Class III) are treated as above (i.e. as being represented by their triggers);
9. Whether or not the negation cue commands the trigger;
10. The parse-tree distance between the event trigger and the negation cue.

We use a default value (null) where none of the other values apply (e.g. when there is no cue in feature 3, 4, 7). These features have been used to train four SVMs on the training dataset: one modelled all events together, and the others modelled the three event classes separately.

## 5 Results

All the results refer to the methods applied on the development dataset (see Table 2). If the negation detection task is regarded as an information extraction task of finding positive instances (i.e.

negated events), then precision, recall, and F-score would be appropriate measures. If we consider the classification aspect of the task, specificity is more appropriate if true negative hits are considered as valuable as true positive ones. We therefore use the following metrics to evaluate the two methods:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = Sensitivity = \frac{TP}{TP+FN}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Specificity = \frac{TN}{TN+FP}$$

where TP denotes the number of true positives (the number of correctly identified negated events), FN is the number of false negatives (the number of negated events that have been reported as affirmative), with TN and FP defined accordingly.

Two sets of negation cues were used in order to compare their influence. A smaller set was derived from related work, whereas additional cues were semi-automatically extracted by exploring the training data. The small negation cue set contains 14 words<sup>3</sup>, whereas the larger negation cue set contains 32 words<sup>4</sup>. As expected, the larger set resulted in increased recall, but decreased precision. However, the effects on the F-score were typically not significant. The results are only shown using the larger cue set.

The texts were processed using the GENIA tagger (Tsuruoka and Tsujii 2005). We used constituency parse trees automatically produced by two different constituency parsers reported in (McClosky et al. 2006) and (Bikel 2004). No major differences were observed in the results using the two parsers. The data shown in the results are produced by the former.

## 5.1 Baseline Results

Our baseline method relies on an implementation of the NegEx algorithm as explained in Section 2.1. Event triggers were used as negation targets for the algorithm. An event is then considered to be negated if the trigger is negated; otherwise it

is affirmative. The results (see Table 3) are substantially lower than those reported for NegEx on clinical data (specificity of 94% and sensitivity of 78%). For comparison, the table also provides an even simpler baseline approach that tags as negated any event whose associated sentence contains any negation cue word.

Approach	P	R	F1	Spec.
any negation cue present	20%	<b>78%</b>	32%	81%
NegEx	<b>36%</b>	37%	<b>36%</b>	<b>93%</b>

Table 3: Baseline results.  
(NegEx and a ‘bag-of-words’ approach)

## 5.2 Rules Based on the Command Relation

Table 4 shows the results of applying the S-command relation rule for negation detection. We experimented with three possible approaches: an event is considered negated if

- the negation cue commands any event participant in the parse tree;
- the negation cue commands the event trigger in the tree;
- the negation cue commands both.

Approach	P	R	F1	Spec.
negation cue commands any participant	23%	<b>76%</b>	<b>35%</b>	84%
negation cue commands trigger	23%	68%	34%	85%
negation cue commands both	23%	68%	35%	<b>86%</b>

Table 4: Performance when only the S-command relation is used.

Compared with the baseline methods, the rules based on the command relation did not improve the performance. While precision was low (23%), recall was high (around 70%), indicating that in the majority of cases there is an S-command relation in particular with the participants (the highest recall). We also note a significant drop in specificity, as many affirmative events have triggers/participants S-commanded by a negation cue (not “linked” to a given event).

## 5.3 Machine Learning Results

All SVM classifiers have been trained on the training dataset using a Python implementation of SVM Light using the linear kernel and the default parameters (Joachims 1999). Table 5 shows the results of the single SVM classifier that has been trained for all three event classes together (applied on the development data).

<sup>3</sup> Negation cues in this set include: *no, not, none, negative, without, absence, fail, fails, failed, failure, cannot, lack, lacking, lacked*.

<sup>4</sup> Negation cues in this set include the smaller set and 18 task-specific words: *inactive, neither, nor, inhibit, unable, blocks, blocking, preventing, prevents, absent, never, unaffected, unchanged, impaired, little, independent, except, and exception*.

Compared to previous methods, there was significant improvement in precision, while recall was relatively low. Still, the overall F-measure was significantly better compared with the rule-based methods (51% vs. 35%).

Feature set	P	R	F1	Spec.
Features 1-7	43%	8%	14%	99.2%
Features 1-8	73%	19%	30%	<b>99.3%</b>
Features 1-9	71%	38%	49%	99.2%
Features 1-10	<b>76%</b>	<b>38%</b>	<b>51%</b>	99.2%

Table 5: The results of the single SVM classifier. Features 1-7 are lexical and POS tag-based features. Feature 8 models whether the cue S-commands any of the participants. Feature 9 is related to the cue S-commanding the trigger. Feature 10 is the parse-tree distance between the cue and trigger.

We first experimented with the effect of different types of feature on the quality of the negation prediction. Table 5 shows the results of the first classifier with an incremental addition of lexical features, parse tree-related features, and finally a combination of those with the command relation between the negation cue and event trigger and participants. It is worth noting that both precision and recall improved as more features are added.

We also separately trained classifiers on the three classes of events (see Table 6). This further increased the performance: compared with the results of the single classifier, the F1 micro-average improved from 51% to 63%, with similar gains for both precision and recall.

Event class	P	R	F1	Spec.
Class I (559 events)	94%	<b>65%</b>	<b>77%</b>	99.8%
Class II (249 events)	<b>100%</b>	33%	50%	<b>100%</b>
Class III (987 events)	81%	44%	57%	99.2%
Micro Average (1,795 events)	88%	49%	63%	99.4%
Macro Average (3 classes)	92%	47%	62%	99.7%

Table 6: The results of the separate classifiers on different classes using common features.

## 6 Discussion

As expected, approaches that focus only on event triggers and their surface distances from negation cues proved inadequate for biomedical scientific articles. Low recall was mainly caused by many

event triggers being too far from the negation cue to be detected as within the scope.

Furthermore, compared to clinical notes, for example, sentences that describe molecular events are significantly more complex. For example, the event-describing sentences in the training data have on average 2.6 event triggers. The number of events per sentence is even higher, as the same trigger can indicate multiple events, sometimes with opposite polarities. Consider for example the sentence

*“We also demonstrate that the IKK complex, but not p90 (rsk), is responsible for the in vivo phosphorylation of I-kappa-B-alpha mediated by the co-activation of PKC and calcineurin.”*

Here, the trigger (phosphorylation) is linked with one affirmative and one negative regulatory event by two different molecules, hence triggering two events of opposite polarities.

These findings, together with previous work, suggested that for any method to effectively detect negations, it should be able to link the negation cue to the specific token, event trigger or entity name in question. Therefore, more complex models are needed to capture the specific structure of the sentence as well as the composition of the interaction and the arrangement of its trigger and participants.

By combining several feature types (lexical, syntactic and semantic), the machine learning approach proved to provide significantly better results. In the incremental feature addition exploration process, adding the cue-commands-participant feature had the greatest effect on the F-score, suggesting the significance of treating event participants. We note, however, that many of the previous attempts focus on event triggers only, although participants do play an important role in the detection of negations in biomedical events and thus should be used as negation targets instead of or in addition to triggers. It is interesting that adding parse-tree distance between the trigger and negation cue improves precision by 5%.

Differences in event classes (in the number and type of participants) proved to be important. Significant improvement in performance was observed when individual classifiers were trained for the three event classes, suggesting that events with different numbers or types of participants are expressed differently in text, at least when negations are considered. Class I events are the simplest (one participant only), so it was expected that negated events in this class would be

the easiest to detect (F-score of 77%). Class II negated events (which can have multiple participants), demonstrated the lowest recall (33%). A likely reason is that the feature set used is not suitable for multi-participant events: for example, feature 8 focuses on the negation cue commanding *any* of the participants, and not *all* of them. It is surprising that negated regulation events (Class III) were not the most difficult to identify, given their complexity.

We applied the negation detection on the type, trigger and participants of pre-identified events in order to explore the complexity of negations, unaffected by automatic named entity recognition, event trigger detection, participant identification, etc. As these steps are typically performed before further characterisation of events, this assumption is not superficial and such information can be used as input to the negation detection module. MacKinlay et al. (2009) also used gold annotations as input for negation detection, and reported precision, recall, and F-score of 68%, 24%, and 36% respectively on the same dataset (compared to 88%, 49% and 63% in our case). The best performing negation detection approach in the BioNLP'09 shared task reported recall of up to 15%, but with overall event detection sensitivity of 33% (Kilicoglu and Bergler 2009) on a 'test' dataset (different from that used in this study). This makes it difficult to directly compare their results to our work, but we can still provide some rough estimates: had all events been correctly identified, their negation detection approach could have reached 45% recall (compared to 49% in our case). With precision of around 50%, their projected F-score, again assuming perfect event identification, could have been in the region of 50% (compared to 63% in our case).

The experiments with rules that were based on the command relations have proven to be generic, providing very high recall (~70%) but with poor precision. Although only the results with S-command relations have been reported here (see Table 4), we examined other types of command relation, namely NP-, PP-, SBAR-, and VP-command. The only variation able to improve prediction accuracy was whether the cue VP-commands any of the participants, with an F-score of 42%, which is higher than the results achieved by the S-command (F-score of 35%). The S-command relation was used in the SVM modules as VP-command did not make the results significantly better.

One of the issues we faced was the management of multi-token and sub-token entities and triggers (e.g. *alpha B1* and *alpha B2* in "alpha B1/alpha B2 ratio", which will be typically tokenised as "alpha", "B1/alpha", and "B2"). In our approach, we considered all the entities that are either multi-token or sub-token. However, if we assign participants that are *both* multi-token and sub-token simultaneously to events and extract similar features for the classifier from them as from simple entities, the F-score is reduced by about 2%. It would be probably better to assign a new category to those participants and add a new value for them specifically in every feature.

## 7 Conclusions

Given the number of published articles, detection of negations is of particular importance for biomedical IE. Here we explored the identification of negated molecular events, given their triggers (to characterise event type) and participants. We considered two approaches:<sup>5</sup> a rule-based approach using constituency parse trees and the command relation to identify negation cues and scopes, and a machine learning method that combines a set of lexical, syntactic and semantic features engineered from the associated sentence. When compared with a regular-expression-based baseline method (NegEx-like), the proposed ML method achieved significantly better results: 63% F-score with 88% precision. The best results were obtained when separate classifiers were trained for each of the three event classes, as differences between them (in the number and type of participants) proved to be important.

The results presented here were obtained by using the 'gold' event annotations as the input. It would be interesting to explore the impact of typically noisy automatic event extraction on negation identification. Furthermore, an immediate future step would be to explore class-specific features (e.g. type of theme and cause for Class III events, and whether the cue S-commands all participants for Class II events). In addition, in the current approach we used constituency parse trees. Our previous attempts to identify molecular events (Sarafraz et al. 2009) as well as those discussed in Section 2 use dependency parse trees. A topic open for future research will be to combine information from both dependency and constituency parse trees as features for detecting negated events.

<sup>5</sup> Available at <http://bit.ly/bzBaUX>

## Acknowledgments

We are grateful to the organisers of BioNLP'09 for providing the annotated data.

## References

- Daniel Bikel. 2004. A Distributional Analysis of a Lexicalized Statistical Parsing. *Proc. Conference on Empirical Methods in Natural Language*.
- Wendy Chapman. 2001. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34(5):301-310.
- Ian Donaldson, Martin, J., de Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G. D., Michalickova, K., Pawson, T. and Hogue, C. W. 2003. PreBIND and Textomy--mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinf.* 4: 11.
- Jörg Hakenberg, Illés Solt, Domonkos Tikk, Luis Tari, Astrid Rheinländer, Quang L. Ngyuen, Graciela Gonzalez and Ulf Leser. 2009. Molecular event extraction from link grammar parse trees. *BioNLP'09: Proceedings of the Workshop on BioNLP*. 86-94.
- Yang Huang and Henry J. Lowe. 2007. A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. *Journal of the American Medical Informatics Association*, 14(3):304-311.
- Thorsten Joachims. 1999. *Making large-Scale SVM Learning Practical*. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.) MIT-Press, MA.
- Halil Kilicoglu, and Sabine Bergler. 2009. Syntactic dependency based heuristics for biological event extraction. *BioNLP'09: Proceedings of the Workshop on BioNLP*. 119-127.
- Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. *BioNLP'09: Proceedings of the Workshop on BioNLP*. 1-9.
- Sofie Van Landeghem, Yvan Saeys, Bernard De Baets and Yves Van de Peer. 2008. Extracting Protein-Protein Interactions from Text using Rich Feature Vectors and Feature Selection. *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine*. 77-84.
- Ronald Langacker. 1969. *On Pronominalization and the Chain of Command*. In D. Reibel and S. Schane (eds.), *Modern Studies in English*, Prentice-Hall, Englewood Cliffs, NJ. 160-186.
- John Lawler. 2010. Negation and Negative Polarity. *The Cambridge Encyclopedia of the Language Sciences*. Patrick Colm Hogan (ed.) Cambridge University Press. Cambridge, UK.
- Andrew MacKinlay, David Martinez and Timothy Baldwin. 2009. Biomedical Event Annotation with CRFs and Precision Grammars. *BioNLP'09: Proceedings of the Workshop on BioNLP*. 77-85.
- James McCawley. 1993. *Everything that Linguists have Always Wanted to Know about Logic But Were Ashamed to Ask*. 2<sup>nd</sup> edition. The University of Chicago Press. Chicago, IL.
- David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-Training for Parsing. *Proceedings of HLT/NAACL 2006*. 152-159.
- Roser Morante and Walter Daelemans. 2009. A Metalearning Approach to Processing the Scope of Negation. *CoNLL '09: Proceedings of the 13th Conference on Computational Natural Language Learning*. 21-29.
- Pradeep Mutalik, Aniruddha Deshpande, and Prakash M. Nadkarni. 2001. Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents: A Quantitative Study using the UMLS. *Journal of the American Medical Informatics Association : JAMIA*. 8(6):598-609.
- Jeyakumar Natarajan, Berrar, D., Dubitzky, W., Hack, C., Zhang, Y., DeSesa, C., Van Brocklyn, J. R. and Bremer, E.G. 2006. Text mining of full-text journal articles combined with gene expression analysis reveals a relationship between sphingosine-1-phosphate and invasiveness of a glioblastoma cell line. *BMC Bioinformatics*. 7: 373.
- Jon Patrick, Yefeng Wang, and Peter Budd. 2006. Automatic Mapping Clinical Notes to Medical Terminologies. *Proc. Of the 2006 Australian Language Technology Workshop*. 75-82.
- Olivia Sanchez. 2007. *Text mining applied to biological texts: beyond the extraction of protein-protein interactions*. PhD Thesis.
- Farzaneh Sarafranz, James Eales, Reza Mohammadi, Jonathan Dickerson, David Robertson and Goran Nenadic. 2009. Biomedical Event Detection using Rules, Conditional Random Fields and Parse Tree Distances. *BioNLP'09: Proceedings of the Workshop on BioNLP*.
- Herman Tolentino, Michael Matters, Wikke Walop, Barbara Law, Wesley Tong, Fang Liu, Paul Fontelo, Katrin Kohl, and Daniel Payne. 2006. Concept Negation in Free Text Components of Vaccine Safety Reports. *AMIA Annual Symposium proc*.
- Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2005. Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data. *Proceedings of HLT/EMNLP 2005*, 467-474.