

CLaC Labs

Processing Modality and Negation

Working Notes for QA4MRE Pilot Task at CLEF 2012

Sabine Rosenberg, Halil Kilicoglu, and Sabine Bergler

CLaC Lab, Concordia University
1455 de Maisonneuve Blvd West, Montréal, QC, Canada, H3W 2B3
sabin_ro@cse.concordia.ca, h_kilico@cse.concordia.ca,
bergler@cse.concordia.ca

Abstract. For the QA4MRE 2012 Pilot Task on Negation and Modality, CLaC Labs implemented a general, lightweight negation and modality module based on linguistic rules. The strong results confirm the suitability of linguistic heuristics for low-level semantic features and showcase their robustness across the different subgenres of the QA4MRE corpora.

Keywords: shallow semantics, negation, modality, factuality

1 Introduction

Processing negation and modality is very relevant for tasks like question answering because extracted information that falls within the scope of a negation cue (*not, no, never, ...*) cannot be presented as asserted information. Similarly, information falling within the scope of a modal cue (*could, should, would, ...*) cannot be presented as factual or certain. Negation and modality have been addressed as independent tasks themselves, rather than as components of other tasks in recent shared tasks and workshops such as detection of negation and hedging at CoNLL [3], BioNLP [6], the Negation and Speculation in NLP Workshop [9] and the *Sem 2012 Shared Task on Processing the Scope and Focus of Negation [10]. The availability of data sets such as the BioScope Corpus [13], and the FactBank Corpus [12] have allowed for further progress on modality and negation research. The QA4MRE Pilot Task: Processing Modality and Negation at CLEF 2012 builds upon these recent efforts.

The Pilot task has 3 main requirements: The first one being to detect whether any elements of a text are negated. Within the context of this task, negation is defined as a grammatical device which is used to determine if an event or situation has not taken place. Consider Example (1), where the event *come* is negated, and one could now infer that half of Europe's electricity comes from other than fossil fuels.

- (1) *Half of Europe's electricity does not come from fossil fuels.*

The second requirement is to detect whether any parts of the text are modalised. For this task: if an event or situation is determined to not be certain nor factual then it is modalised. In Example (2), the event *come* is considered to be modalised due to the auxiliary *might* which renders the event to be uncertain - it is neither true nor false that half of Europe’s electricity comes from fossil fuels - but possibly it does.

(2) *Half of Europe’s electricity might come from fossil fuels.*

The third requirement of the task is the most complex: to determine, whether and how the two cases might interact. Example (3) demonstrates that the event *come* is modalised as in Example (2), however it is also negated as in Example (1). The interpretation of this sentence is completely different due to both negation and modality affecting the event *come* - it is not a certain fact that half of Europe’s electricity comes from other than fossil fuels - but it possibly does.

(3) *Half of Europe’s electricity might not come from fossil fuels.*

The pilot task is framed as an annotation task where each document in the data set is pre-annotated for events. An event is defined as any of the main verbs mentioned within the text. The events are extracted automatically with the Stanford POS Tagger (v. 3.0, 2010-05-10) [8]. The goal of the task is to then determine whether an event present in the text is within the scope of a negation, in the scope of a modal or within the scope of both.

We present a linguistically inspired general Negation and Modality detection system that identifies negation and modality triggers, and in a second step determines their scope from dependency graphs. In the final step, the only one specific to the pilot task, we determine whether any mentioned event within the text is within the detected scopes of negation, modality or within the intersection of both. Consequently, the output of the system is a list of all the events, each one with a specific label - the possible labels are NEG: the event is found to be negated only, MOD: the event is found to be modalised only, NEGMOD: the event is found to be both negated and modalised, or NONE: the event is found neither to be negated nor modalised.

2 The Data Set

The test data set for the pilot task is composed of 8 English documents part of the larger test set for the main QA4MRE Task. It is divided into 4 topics: AIDS, Climate Change, Music and Society, and Alzheimer’s Disease. Two documents were taken from each topic. Each test document has at least 100 events (on average) to which the relevant label should be assigned. The training set consisted of one document from the Economist. It contained 96 events. 46 of the events in the training example were labeled as MOD, 49 were labeled as NONE, 1 was labeled as NEG and 0 were labeled as NEGMOD. The guidelines for the pilot task [8] also contained some extra example sentences for each possible label - which proved helpful in developing our system. We also utilized the background collection data available for the QA4MRE task in order to develop our heuristics.

3 Our Approach

For the pilot task, CLaC Labs implemented a general lightweight Negation and Modality module. The heuristics-based system is composed of three modules for the GATE [1] environment: the first component detects and annotates negation and modality cues present in the corpus, the second component detects and annotates the syntactic scope of the detected instances of negation and modality. We initially developed these first two components for specifically detecting and annotating instances of negation within text. This base system NEGATOR was developed independently drawing on data from MPQA [14] and TIMEBANK[11] with validation on Bio-Scope [13], and the *SEM 2012 Task [10] on detecting the focus of Negation. We extended NEGATOR for the requirements of the Pilot task to include the functionality for also detecting instances of modality within text. The third component was specifically developed for the pilot task and is intended to validate the first two, now extended components and for the determination of which of the annotated events are within the detected scopes of negation, modality or within the intersection of both.

3.1 Data Preprocessing

Parser-based, our modality and negation detection pipeline requires as input entire sentences. The system then performs standard preprocessing using prebuilt GATE modules [1]: sentence splitting, tokenization, parsing using the Stanford Parser [7, 2], morphological preprocessing, and VP chunking. In the final preprocessing step, the system extracts the pre-annotated events from the text already present in the supplied corpus for the task in order for them to be available in the final step of the pipeline.

3.2 Detection of Negation and Modality Triggers

The first component in the CLaC System identifies and annotates any negation or modality triggers present in the text. For the detection of negation triggers we used 3 different xml formatted gazetteer lists. Each list contains entries for a specific negation type: Implicit, Explicit and Affixal. The Explicit trigger list includes *not, never, no, don't, can't, ...* The Implicit trigger list includes *deny, prevent, fail, reject, ...* Affixal triggers includes *disagree, unimportant, impossible, incoherence, ...* Both the Affixal and Implicit gazetteer lists are seeded from the subjective word list from MPQA, complemented with words annotated in MPQA [14] and nominalization/verbal form variants. For the detection of modality triggers we used a comprehensive modality dictionary developed and compiled for [5]. From the possible modals we chose the following categories according to the task guidelines [8]: Assumptive, Conditional, Deductive, Epistemic, Hedge, Intentional, Speculative, Obligative, Modal, Potential, and Volitive.

Two additional modality trigger classes are (the trigger is underlined):

1. if the main verb is nonfinite then this verb is considered to be a modal trigger: ... *that accompany Alzheimer's memory loss ...*

2. if the main verb is in the present participle tense then this verb is considered to be a modal trigger: *Finding evidence of Chlamydia...*

3.3 Syntactic Scope Detection of Negation and Modality

The second component in the CLaC System identifies the syntactic scope for an annotated negation or modal trigger present in a sentence within the text. The method for detecting the syntactic scope in both cases is inspired by heuristics defined in [4], utilizing the parser-derived collapsed dependency graphs [2]. The heuristics specify the dependency relations which identify either a negation or modality scope. Examples (4) and (5) from the test dataset show how the resulting scope annotations (in angle brackets) are determined.

In Example (4) the CLaC System will first detect and annotate *never* as an Explicit Negation Trigger. The CLaC component for identifying the syntactic scope will then trigger the heuristic for the **neg** dependency relation as the Explicit Negation Trigger *never* is a member of an instance of this relation. The resulting syntactic scope is determined to be the VP headed by the verb *happened*. The verb *happened* is consequently also marked as an event, therefore the resulting label for this event will be NEG since *happened* is found by the system to be within the scope of a negation.

- (4) **Sentence:** *The cut never happened.*
Explicit Negation Trigger: never
Dependency Relation: neg(governor: happened, dependent: never)
Negation Syntactic Scope: *The cut never <happened>.*
System Labeled Event: happened : LABEL =NEG

The same process occurs in Example (5): The system will first detect and annotate *can* as a Modality Trigger. It will then trigger the heuristic for the aux dependency relation as the Modality Trigger *can* is a member of an instance of this relation. The scope is the VP headed by the verb *change*. The verb *change* is also marked as an event, therefore the resulting label for this event will be MOD since *changed* is found by the system to be within the scope of a modal.

- (5) **Sentence:** *But it is hardly evidence that hip-hop can change the world.*
Modality Trigger: can
Dependency Relation: aux(governor: change, dependent: can)
Modality Syntactic Scope: *But it is hardly evidence that hip-hop can <change the world>.*
System Labeled Event: change : LABEL =MOD

There are two trigger classes that do not use the dependency relations to determine the syntactic scope, but rather the constituent parse tree, because it more accurately reflects the extent of the scope:

1. Modal verbal triggers not marked by a dependent relation already (the syntactic scope is the VP complement: *accompany Alzheimer's memory loss*

2. not only, not just, are exception cases and the scope of the explicit negation trigger is limited to: (*only, just*).

3.4 Determination of Events: Modalised, Negated or both

The final component implemented for the pilot task iterates through all the pre-annotated events within the text and will output the following labels:

1. **NEG**: if the event is determined to be negated
2. **MOD**: if the event is determined to be modalised
3. **NEGMOD**: if the event is determined to be both modalised and negated
4. **NONE**: if the event is neither negated nor modalised

For this task we submitted two different runs. The only difference between the two is how the final component of the system determines the label for the event. In the 1st run known as the "Narrow" run one mainly considers the direct members of the dependency relation that triggered a relevant scope annotation. In contrast, the 2nd run, known as the "Greedy" run, the entire scope annotation in which an event is contained within is considered.

Run 1: Narrow

1. The event is determined to be NEG only if it is contained within a negation scope annotation, and is a direct member in the dependency relation with the negation trigger as shown in Example 6. If the event is not a direct member, but the event is contained within a negation scope annotation, and has the infinitive form *to be*, and is the dependent in a copula dependency relation and the governor of the copula dependency relation is a direct member in the dependency relation with the negation trigger: then this event will also be given the NEG label.
 - (6) **Sentence:** *The mutation means that Duffy receptor proteins are not made in red cells .*
Negation Trigger: not
Dependency Relation: neg (governor:made, dependent: not)
Negation Scope: *The mutation means that Duffy receptor proteins are not (made in red cells).*
System Labeled Event: made : LABEL =NEG
2. The event is determined to be MOD only if it is contained within a modal scope annotation, and is a direct member in the dependency relation with the modal trigger. If the event is not a direct member, but the event is embedded in certain structural constructions, the event will also be given the MOD label as seen in Example 7.
 - (7) **Sentence:** *Sometimes the absence of something can be as telling as its presence .*
Modal Trigger: can
Dependency Relation: aux (governor: telling, dependent: can)

Dependency Relation: cop (governor: telling, dependent: be)
Modal Scope: *Sometimes the absence of something can (be as telling as its presence) .*
System Labeled Event: be : LABEL =MOD

3. The event is determined to be NEGMOD only if it has been allocated both the preliminary MOD and NEG labels as shown in Example 8.

(8) **Sentence:** *So you can't simply look at temperature changes over the 20th century ...*
Modal Trigger: can
Dependency Relation: aux(governor: look, dependent: ca)
Preliminary System Labeled Event: look : LABEL =MOD
Modal Scope: *So you ca (n't simply look at temperature changes over the 20th century) ...*
Negation Trigger: n't
Dependency Relation: neg (governor: look, dependent: n't)
Negation Scope: *So you ca n't (simply look at temperature changes over the 20th century) ...*
Preliminary System Labeled Event II: look : LABEL =NEG
Final System Labeled Event: look : LABEL =NEGMOD

4. If none of the above is true then the event is labeled as NONE.

Run 2: Greedy

1. The event is determined to be NEG if it is contained within a negation scope annotation as shown in Example 9.

it would be better if the same example could showcase narrow and greedy scope to highlight the difference

(9) **Sentence:** *Nor is it as widespread in Africa*
Negation Trigger: Nor
Dependency Relation: dep (governor: widespread, dependent: Nor)
Negation Scope: *Nor (is it as widespread in Africa)*
System Labeled Event: is : LABEL =NEG

2. The event is determined to be MOD if it is contained within a modal scope annotation as shown in Example 10. If the scope was determined not by a dependency relation but by a structural trigger, then this event is labeled MOD.

(10) **Sentence:** *If matter and antimatter were truly symmetrical...*
Modal Trigger: If
Dependency Relation: mark (governor: symmetrical, dependent: if)
Modal Scope: *If (matter and antimatter were truly symmetrical)...*
System Labeled Event: were : LABEL =MOD

3. The event is determined to be NEGMOD only if it has been allocated both the preliminary MOD and NEG labels as shown in Example 11.

- (11) **Sentence:** *This would probably not be a person you would take very seriously*
Modal Trigger: would (2nd one)
Dependency Relation: aux(governor: take, dependent: would)
Modal Scope: *This would probably not be a person you would (take very seriously) .*
Preliminary System Labeled Event: take : LABEL =MOD
Negation Trigger: not
Dependency Relation: neg (governor: person, dependent: not)
Negation Scope: *This would probably not (be a person you would take very seriously).*
Preliminary System Labeled Event II: take : LABEL =NEG
Final System Labeled Event: take : LABEL =NEGMOD

4. If none of the above is true then the event is labeled as NONE.

4 Global Results

Two methods are used for calculating the performance of a system based on the precision and recall measures: the Macroaverage and the Microaverage.

The Microaverage F-Score will be the harmonic mean of the Microaverage precision and recall. In contrast, the Macroaverage method will calculate the Precision by simply taking the average of the Precision values calculated individually for each category. Similarly to calculate the Recall, it will take the average of the Recall values. The Macroaverage F-Score will be the harmonic mean of the Macroaverage precision and recall, shown in Table 1.

	Narrow Greedy	
Macroaveraged F-measure(beta=1.0):	0.6368	0.6196
Microaveraged F-measure(beta=1.0):	0.7117	0.6750
Overall Accuracy:	0.7130	0.6688

Table 1. Averaged Scores for Narrow and Greedy Run

We observe that the Macroaveraged measure is balanced across both runs with 64% for the Narrow run and 62% for the Greedy run. The Narrow run had better overall performance. The method for allocating the labels in the Narrow run was not overly generous in the MOD/NEG/NEGMOD labeling of the events. Consequently, we observe in Table 2 the Narrow run performs better in detecting the NONE label correctly for an event. The detection of the NEGMOD label for an event is the worst performer for both runs. This is possibly due to the manner in which the NEGMOD label is allocated: the correct determination of an event being labeled NEGMOD is reliant on it having been previously correctly allocated the NEG label and the MOD label. Any errors that occur in the allocation of the NEG or MOD label to an event will propagate through

Narrow Run							
	TP	FP	TN	FN	Precision	Recall	F-Measure(beta =1.0)
MOD	315	158	612	159	0.6660	0.6646	0.6653
NEG	36	10	1170	28	0.7826	0.5625	0.6545
NEGMOD	18	19	1184	23	0.4865	0.4390	0.4615
NONE	518	170	409	147	0.7529	0.7789	0.7657
Greedy Run							
	TP	FP	TN	FN	Precision	Recall	F-Measure(beta =1.0)
MOD	357	252	518	117	0.5862	0.7532	0.6593
NEG	36	8	1172	28	0.8182	0.5625	0.6667
NEGMOD	28	55	1148	13	0.3373	0.6829	0.4516
NONE	411	97	482	254	0.8091	0.6180	0.7008

Table 2. Global Results for Narrow and Greedy Run

to the NEGMOD labeling task. In both runs the F-Measures for both the NEG and MOD labels are fairly stable: 65%-66%.

5 Error Analysis

This section reviews the results of each run from a more fine grained analysis than the previous section. Here, we investigate how each document in the provided data set performs, and identify the major sources of errors. We approach the analysis of the results at two different levels of granularity.

The averaged performance of the Narrow and Greedy run for each document in the data set is depicted in Figures 1 and 2.

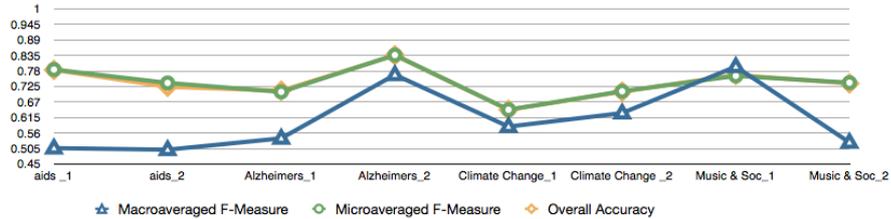


Fig. 1. Narrow Run: Averaged Scores for each test document

The macroaveraged scores range between 50%- 80% for the Narrow run and 50%- 76% for the Greedy run. Given the very small size but very varied nature of the test corpus, it is of interest to us to analyse where a particular document falls within the given range. The worst performing document overall is the *aids_2* document for both runs: 50.11% for the Narrow run and 49.24% for the Greedy run. The best performing document in the Narrow run is *Music & Soc_1* with

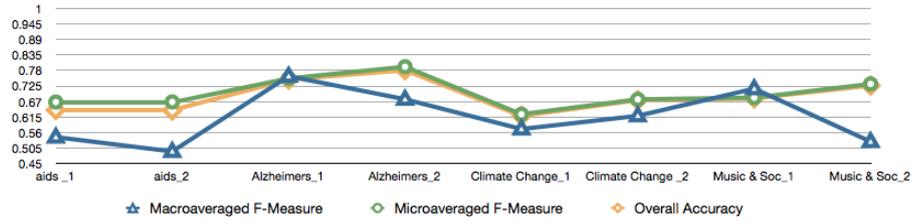


Fig. 2. Greedy Run: Averaged Scores for each test document

79.55% and *Alzheimers_2* second with 76.66%. In contrast, the best performer in the Greedy run is *Alzheimers_1* with 75.95% and *Music & Soc_1* second with 71.54% .

The Figures 3 & 4 depicts a further level of granularity on the results for each document in the data set.

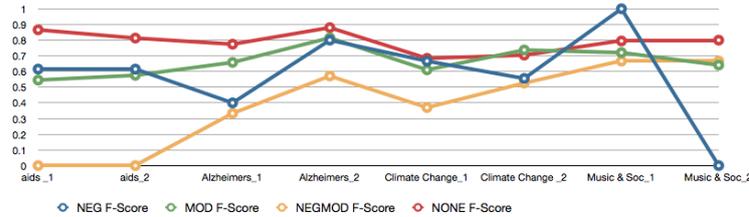


Fig. 3. Narrow Run: F-Measures of NEG, MOD, NEGMOD and NONE categories

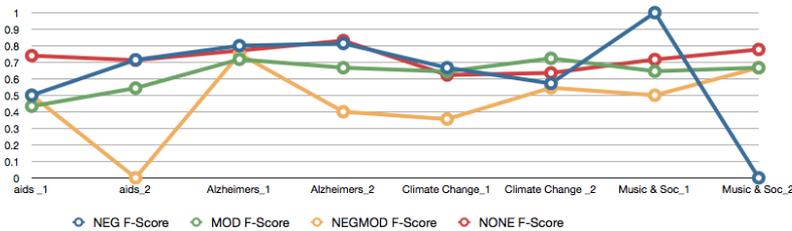


Fig. 4. Greedy Run: F-Measures of NEG, MOD, NEGMOD and NONE categories

Upon inspection of these figures we see that in both runs the *aids_2* does poorly in detecting the NEGMOD label. There is only one event in this document that should have the NEGMOD label allocated; and in both runs the system fails. In both runs as observed in Example 12, the event is labeled as MOD but

the system does not detect a negation trigger and therefore the event is not allocated the NEG label:

- (12) **Sentence:** *That is two years when a man ignorant of his infection would be less likely to take precautions to stop it spreading.*
ModalTrigger: take
Modal Scope: *That is two years when a man ignorant of his infection would be less likely to take (precautions to stop it spreading).*
System Labeled Event: take : LABEL =MOD
Gold Labeled Event: take : LABEL =NEGMOD

The *aids_2* document is not the worst performer in either run for detection of the MOD label. In both runs the worst performer when detecting the MOD label is the *aids_1* document. This poor performance is due in the majority of cases to a mislabeling of the MOD label: the system labels them MOD and the Gold Standard allocates the NONE label. This mislabeling is relatively frequent and we illustrate it with a few examples:

1. The CLaC system detects a modal trigger that the Gold Standard does not:

(13) **Sentence:** *The fact that some matter was left over shows they are not , in fact , symmetrical.*
ModalTrigger: fact
Modal Scope: *The fact (that some matter was left over) shows they are not , in fact , symmetrical.*
System Labeled Event: left : LABEL =MOD
Gold Labeled Event: left : LABEL =NONE
2. The Gold Standard considers the event to not be affected by the modal context, but the Greedy Run does:

(14) **Sentence:** *A tiny genetic change may help explain why AIDS is so common in Africa.*
ModalTrigger: may
Modal Scope: *A tiny genetic change may (help explain why AIDS is so common in Africa) .*
System Labeled Event: explain : LABEL =MOD
Gold Labeled Event: explain : LABEL =NONE
3. The ClaC System wrongly allocates a term to be a modal trigger - the term may be a modal trigger but in a different context:

(15) **Sentence:** *Dr Gallo had initially suggested that AIDS was caused by HTLV-I, a virus that no one disputes he discovered.*
ModalTrigger: had
Modal Scope: *Dr Gallo had (initially suggested that AIDS was caused by HTLV-I, a virus that no one disputes he discovered.)*
System Labeled Event: suggested : LABEL =MOD
Gold Labeled Event: suggested : LABEL =NONE

The worst performer of the NEG label for both the Narrow and Greedy run is *Music & Soc_2*. This is in part due to the small number of NEG events in comparison to the other documents. The second worst performer of the NEG

label in the Narrow run is *Alzheimers_1* and in the Greedy Run it is *aids_1*. The *Alzheimers_1* was quite a good NEG Performer in the Greedy Run (2nd best). Of interest is that in the *Alzheimers_1* document the Greedy run holds the advantage. As seen in Example 16, the event in question is not directly related to the negation trigger through the relevant dependency relation:

- (16) **Sentence:** *Nor does the apparent correlation with Alzheimer 's prove anything.*
NegTrigger: Nor
Negation Scope: *Nor <does the apparent correlation with Alzheimer's prove anything>.*
System Labeled Event: prove : LABEL =NEG
Gold Labeled Event: prove : LABEL =NEG

Error cases illustrating NEG labeling errors from the *aids_1* document:

1. The CLaC System erroneously allocates NEG and MOD labels:

(17) **Sentence:** *Dr Gallo had initially suggested that AIDS was caused by HTLV-I, a virus that no one disputes he discovered.*
ModalTrigger: suggested
Modal Scope: *Dr Gallo had initially suggested <that AIDS was caused by HTLV-I, a virus that no one disputes he discovered>.*
NegTrigger: no
Negation Scope: *Dr Gallo had initially suggested that AIDS was caused by HTLV-I, a virus that no<one disputes he discovered>.*
System Labeled Event: disputes : LABEL =NEGMOD
Gold Labeled Event: disputes : LABEL =NEG
2. The CLaC System does not recognise the Negation Scope:

(18) **Sentence:** *What the committee did not do was name a third HIV researcher, Robert Gallo, to share the glory and the SKr 10m (\$1.4m).*
NegTrigger: not
ModTrigger: share
Negation Scope: *What the committee did not <do was name a third HIV researcher,> Robert Gallo, to share the glory and the SKr 10m (\$1.4m).*
System Labeled Event: share : LABEL =MOD
Gold Labeled Event: share : LABEL =NEG

6 Conclusion

The QA4MRE Pilot Task on Negation and Modality has added crucial data to the analysis not only of the two targeted linguistic phenomena, but indeed to the study of interactions between modules in the light of increased complexity.

CLaC Labs converted prototype systems to detect negation and modality that were combined for this Shared Task. We attribute our strong results to the careful linguistic analysis that went into the component systems. The Shared Task confirmed our pretheoretic assumptions that for low-level semantic features, linguistic rules will trump statistical methods for some time to come, since

the amount of coherently annotated data is not available for such approaches to excel. We have also been able to show that these general (and very simple) linguistic rules perform similar in the different subgenres of the QA4MRE corpora, an encouraging result. The data set for the shared task is insufficient to make any stronger claims, but we certainly feel encouraged to extend our current approach to more interacting modules. The careful and explicit study of the interactions will, in our opinion, shed light on complexity issues in general, well beyond the task issues annotated.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M., Saggion, H., Petrak, J., Li, Y., P., W.: Text Processing with GATE (Version 6). GATE (April 15, 2011) (2011)
2. de Marneffe, M., Manning, C.: Generating typed dependency parses from phrase structure parses. In: LREC. (2006)
3. Farkas, R., Vincze, V., G.Móra, Csirik, J., G.Szarvas: The conll-2010 shared task: Learning to detect hedges and their scope in natural language text. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning. (2010)
4. Kilicoglu, H., Bergler, S.: Effective bio-event extraction using trigger words and syntactic dependencies. *Computational Intelligence* **27**(4) (2011) 583–609
5. Kilicoglu, H.H.: Embedding Predications. PhD thesis, Concordia University, Montreal, Quebec (2012)
6. Kim, J.D., Wang, Y., Takagi, T., Yonezawa, A.: Overview of genia event task in bionlp shared task 2011. In: Proceedings of BioNLP Shared Task 2011 Workshop at ACL-HLT. (2011)
7. Klein, D., Manning, C.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. (2003)
8. Morante, R., Daelemans, W.: Processing modality and negation. a pilot task of the qa4mre lab, CLEF 2012 Notebook papers (2012)
9. Morante, R., Sporleder, C., eds.: NeSp-NLP '10: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, Stroudsburg, PA, USA, Association for Computational Linguistics (2010)
10. Morante, R., Blanco, E.: *SEM 2012 Shared Task: Resolving the Scope and Focus of Negation. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM 2012), Montréal, Canada (June 2012) 265–274
11. Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L. and Lazo, M.: The timebank corpus. In: Proceedings of Corpus Linguistics. (2003)
12. Sauri, R., Pustejovsky, J.: Factbank: a corpus annotated with event factuality. *Language Resources and Evaluation* **43**(3) (2009) 227–268
13. Vincze, V., Szarvas, G., Farkas, R., Mra, G., Csirik, J.: The bioscope corpus: annotation for negation, uncertainty and their scope in biomedical texts. *BMC Bioinformatics* **9**(Suppl 11):S9 (2008)
14. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* **39**(2-3) (2005)