

# The Influence of Syntactic Information on Hedge Scope Detection

Guillermo Moncecchi<sup>1</sup>, Jean-Luc Minel<sup>2</sup>, and Dina Wonsever<sup>1</sup>

<sup>1</sup> Instituto de Computación, Facultad de Ingeniería  
Universidad de la República, Montevideo, Uruguay

<sup>2</sup> Laboratoire MoDyCo, UMR 7114 CNRS  
Université Paris Ouest Nanterre La Défense, France

**Abstract.** In this paper we elaborate over the use of sequential supervised learning methods on the task of hedge cue scope detection. We address the task using a learning methodology that proposes the use of an iterative, error-based approach to improve classification performance. We analyze how the incorporation of syntactic constituent information to the learning and post-processing steps produces a performance improvement of almost twelve points in terms of F-score over previously unseen data.

## 1 Introduction

The use of speculative language poses an interesting problem for Natural Language Processing (NLP), since it potentially reflects the subjective position of the writer towards the truth value of certain facts; when the fact is extracted or used for inference, this certainty information should not be lost. *Hedging*, a term first introduced by Lakoff (1973) to describe the use of ‘words whose job is to make things fuzzier or less fuzzy’ is ‘the expression of tentativeness and possibility in language use’ (Hyland, 1995), and is extensively used in scientific writing. Several authors have studied the phenomenon of speculation from philosophical, logical and linguistic points of view, generating a rich and heterogeneous theory about speculative language. All this body of work should aid in the construction of an NLP system for speculative sentence detection.

The most common approach in the NLP literature (see, for example, Farkas et al. (2010)) to speculative language detection within scientific writing is through hedge detection: the presence of lexical tentativeness or possibility marks (‘hedge cues’) within a sentence is considered an indicator of speculation. The linguistic devices used to express hedging are diverse: besides modal verbs, epistemic lexical verbs (such as ‘suggest’ or ‘indicate that’), adjectives (‘likely’), adverbs (‘probably’), and even nouns (‘possibility’) are frequently used to express uncertainty. For example, in the following sentence, the modal verb ‘may’ is epistemically used to avoid bald assertion:

*Example 1.* Since clinical remission has been observed in a significant fraction of DLCL cases, these markers may serve as critical tools for sensitive monitoring of minimal residual disease and early diagnosis of relapse.

The sentence expresses speculation, and the hedge ‘may’ serves to show the possibility of a certain procedure. But there is another fact within the sentence (that clinical remission have been observed in many cases) which is not hedged, so marking the whole sentence as speculative could lead to the wrong assumption that this fact is merely a possibility. The notion of *hedge scope* (Morante et al., 2009; Vincze et al., 2008) captures the idea that it is possible that only a fragment of a sentence is hedged. In the previous example, we could consider that the scope of the ‘may’ hedge cue could be the verb phrase that starts with the very word<sup>3</sup>:

*Example 2.* Since clinical remission has been observed in a significant fraction of DLCL cases, these markers {may serve as critical tools for sensitive monitoring of minimal residual disease and early diagnosis of relapse}.

Hedges, when lexically marked, can be considered linguistic operators, therefore inducing a scope. However, the very notion of hedge scope is introduced for the Bioscope corpus annotation (Vincze, 2008), and is never formally defined, presenting instead a series of criteria to identify them, based mainly on syntax. We have previously suggested (Moncecchi et al., 2013) defining hedge scope as *the part of meaning in the sentences that is hedged*, i.e. where the tentativeness or possibility holds. From this perspective, scope is a *semantic* notion, that can be, however, strongly related with syntax.

In this paper we show how we applied supervised sequential learning methods to the task of scope detection. In particular, we want to analyze how the incorporation of different linguistic information (including lexical and syntactic information) to the learning and post-processing steps allowed to improve prediction performance<sup>4</sup>. To achieve this improvement we propose an iterative methodology, where errors committed on a held out corpus are inspected and general rules (mostly based on expert knowledge on syntax) are proposed to solve them. These rules are incorporated as features for the learning process.

In the next section we review previous work on hedge cue identification and scope detection, with special emphasis on this second task. Section 3 presents the learning methodology. In section 4 we analyze performance improvement after incorporating diverse learning attributes, mainly from syntactic scope information, and discuss when these improvements hold on previously unseen data. Finally, we analyze the presented approach, suggesting future lines of research.

## 2 Related Work

The first approaches to speculation detection aimed to classify each sentence as speculative or not speculative. Medlock and Briscoe (2007), for example, pro-

<sup>3</sup> In this work, hedge cues will be underlined and their scope marked with brackets, annotated with the name of the hedge cue in case there are multiply nested scopes

<sup>4</sup> Obviously, for scope to be detected we have first to identify sentence hedge cues; in this paper (for space reasons), we assume that this step is completed, and use gold-standard hedge cues, concentrating on the problem of scope detection

posed to use a semi-supervised learning approach to solve this binary classification task. Morante et al. (2009), adapting previous work on negation detection, reduced speculation detection to hedge cue identification: a sentence would be considered speculative if it included one or more hedge cues.

For the scope detection task, Morante et al. (2009) proposed to address it as a case of *sequential classification*: given a sentence and a hedge cue within it, classify each sentence token, indicating if it is part of the hedge scope, using a so-called FOL marking: assign class F to the first token of the predicted scope, and L the last one, marking every other token with class O. The classifier input included both the sentence and the hedge cue since a sentence could include two or more cues. For this task, only those sentences where a hedge cue had been found were used as learning instances.

Several works have been presented using this same approach, specially during the CoNLL-2010 Shared Task (Farkas et al., 2010), ranging from pure rule-based studies, based on sentence syntactic structure (Ozgur et al., 2009; Ovreliid et al., 2010; Velldal et al., 2010), to machine learning and hybrid systems (Morante et al., 2010), incorporating lexical and syntactical information. However, the task of scope resolution is still far from resolved: the best reported result for the CoNLL evaluation corpus is a F-measure of 0.696 (Velldal et al., 2012), using gold-standard hedge cues.

### 3 Methodology

To address the sequential learning task of scope detection, we followed a learning methodology that starts with an initial guess of attributes for supervised learning and a learning method, and builds a classifier; this classifier is in turn evaluated on a held out corpus, and errors are handled to an expert, who proposed rules for solving, based on present or proposed lexical, positional or syntactic attributes. In this section we briefly explain the process. For a detailed explanation see Moncecchi et al. (2010, 2012, 2013).

We used the abstracts section of the Bioscope corpus (Vincze et al., 2008) as a learning source and for evaluation purposes. We first divided the corpus into a 80% training corpus and a 20% evaluation corpus; since we were using an iterative methodology, we further separated a 10% held out corpus from the training set, and used it for tuning.

Following previous studies, we addressed scope detection as a sequential learning task, using FOL classes for each token. Table 1 shows a scope learning instance for the following sentence:

*Example 3.* This finding {suggests that the BZLF1 promoter may be regulated by the degree of squamous differentiation}.

The input format is the standard learning format used in the CoNLL Shared Tasks (Buchholz, 2006). For each hedge cue in a sentence, a learning instance is generated. In the previous example, the hedge cue is the word ‘suggests’ (you can see it marked with a B value for the ‘Hedge cue’ attribute in Tables 1-2), while

its scope is the text span between tokens 3 and 16. There is another, different, learning instance coming from the same sentence, originated by the hedge cue ‘may’ spanning from tokens 5 to 16 (according to the Bioscope corpus annotation guidelines). Each learning instance could include an arbitrary number of additional attributes, resulting, for example, from lexical or syntactical analysis.

To evaluate classifier performance, we took a perfect-match approach: we considered an evaluation instance as correctly identified only if every token in the sentence were correctly classified. This means that, for the case of scope detection, the scope was considered correctly detected if both the first and last token of the scope were correctly marked. Classification performance was measured in terms of the traditional figures of precision, recall, and F-score. For the scope detection task, these three numbers coincided, since every False Positive (instances with incorrectly classified scope), implied a False Negative (instances where the correct scope was not identified).

The sequential learning process acts as expected: from the learning instances, a model is constructed using a learning method. We used linear-chain Conditional Random Fields, a statistical method, since it had produced state-of-the-art result in many sequential learning tasks (Lafferty, 2001). Besides the usual lexical or syntactical attributes related to each word (e.g. POS tags, lemmas, syntax tree parent), we introduced a special type of attributes, which we called *knowledge rules*. Their name comes from the fact that they try to model a prediction on the target classes, given an instance, product of an expert rule. For example, a rule that states ‘the scope of the keyword ‘may’ is the same as the scope of the parent node of the hedge cue in the syntax tree’, would be represented by an attribute (with exactly the same possible values as the target class), whose F and L tokens will be, respectively, the first and last token of the syntax scope of the parent node of ‘may’. Note that these knowledge rules are *not* post-processing rules, but just learning attributes. The learning method will decide if they are relevant for learning, just as any other attribute. We will see some examples later in this article.

In the next section we show how we defined a base classifier and incrementally improved it. Since our procedure is based on analyzing errors, we used a held out corpus to evaluate improvements; classifier measurement on the evaluation corpus was only done after finishing the iterative process. At the end of the section, we will show how each classifier performed on the evaluation corpus, so as to measure if the improvements on the held out corpus hold.

## 4 Classification Improvement

### 4.1 Baseline Classifier

What we considered as learning attributes for our baseline classifier were the same lexical attributes (word, lemma, and part-of-speech class<sup>5</sup>) we used for

<sup>5</sup> We applied the GENIA tagger (Tsuruoka et al., 2005) to the original corpus to get the lexical information for each word, and the Stanford Parser (Klein et al., 2003) to get each sentence syntax tree

hedge cue identification, adding the hedge cue attribute learned in the previous step; all four attributes were considered in a window of size 5, centered in the current token. Figure 1 shows the learning instance for this classifier, for our previous example. This classifier also includes a set of post-processing rules that are fired when one (or both) scope limits could not be identified: when we transform the task of scope detection into a sequential classification one and use a FOL format, it is possible that not exactly one F and one L class are predicted for an instance. In this case, we know for sure that the correct scope could not be identified, and should try to back off to a simpler guess. The set of rules we used to modify the classifier results on evaluation data to correct those cases were a slightly modified form of the rules presented by Morante et al. (2009).

This baseline classifier, including the post-processing rules, obtained an F-score of 0.66 on the held out corpus.

Token	Word	Lemma	POS	Hedge cue	Scope
1	This	This	DT	O	O
2	finding	finding	NN	O	O
3	suggests	suggest	VBZ	B	F
4	that	that	IN	O	O
5	the	the	DT	O	O
6	BZLF1	BZLF1	NN	O	O
7	promoter	promoter	NN	O	O
8	may	may	MD	O	O
9	be	be	VB	O	O
10	regulated	regulate	VBN	O	O
11	by	by	IN	O	O
12	the	the	DT	O	O
13	degree	degree	NN	O	O
14	of	of	IN	O	O
15	squamous	squamous	JJ	O	O
16	differentiation	differentiation	NN	O	L
17	.	.	.	O	O

**Table 1.** Base classifier attributes

## 4.2 Iteration 1: Adding Syntax Information

We observed that most scopes were associated with the sentence syntactic constituents, particularly those that included the hedge cue (this was reasonable, since the corpus annotation guidelines were mostly based on syntax). For example, the scope of a lexical verb such as ‘suggest’ matched the parent component of the hedge cue in the parse tree, as the parse tree in Figure 1 shows:

To improve classification, we included as a learning attribute a knowledge rule stating that the scope of the hedge cue was the syntactic scope of the parent of the hedge cue (i.e. the parent in the parse tree of its first word), modulo final periods. Or, in terms of attribute values:

```

(ROOT
 (S
  (NP (DT This) (NN finding))
  (VP (VEZ suggests)
   (SBAR (IN that)
    (S
     (NP (DT the) (NN BZLF1) (NN promoter))
     (VP (MD may)
      (VP (VB be)
       (VP (VBN regulated)
        (PP (IN by)
         (NP
          (NP (DT the) (NN degree))
          (PP (IN of)
           (NP (JJ squamous) (NN differentiation))))))))))
    (. )))

```

**Fig. 1.** Parsing tree for sentence 3. The scope of the hedge cue is shown in bold

`in_hc_parent_scope` = F if the token is the first word of the parent of the HC  
= L if the token is the last word of the parent of the HC  
= 0 otherwise

Since this criterion does not hold for every part-of-speech (and not even for every use of the verb, as, for example, passive voice construction or raising verbs cases show), we also included as a learning attribute the part-of-speech of the hedge cue parent. Table 2 shows the classifier attributes for the ‘suggest’ hedge cue in sentence 3. We can see that the `in_hc_parent_scope` rule matches the cue scope; that does not happen in the case of the hedge cue ‘may’, where the syntactic scope starts with the hedge cue, while the hedge scope starts with token 5, according to annotation guidelines. We expected that the classifier could confirm or discard a correlation between the knowledge rule and the scope, based on training data, as it does with every other attribute.

After a grid search for the best attributes and fine-tuning learning parameters (w.r.t performance on the held-out corpus), we obtained a classifier with an F-measure of 0.705 on the held out corpus, improving by four points with respect to the base classifier.

### 4.3 Iteration 2: Adding Ancestors in the Syntax Tree

After the previous iteration, we elaborated a list of the 116 errors the classifier had committed in the held-out corpus and tried to understand why it had been wrong. As we have already said, this method could be applied because attributes for learning corresponded quite directly with a few observable properties related to the linguistic phenomenon we aimed to study.

Analyzing these errors, we found that assuming that only parent scopes were related with hedge scopes was not enough. In some cases, such as those of passive

Token	Word	Lemma	POS	HC	PPOS	in-PScope	Scope
1	This	This	DT	O	VP	O	O
2	finding	finding	NN	O	VP	O	O
3	suggests	suggest	VBZ	B	VP	F	F
4	that	that	IN	O	VP	O	O
5	the	the	DT	O	VP	O	O
6	BZLF1	BZLF1	NN	O	VP	O	O
7	promoter	promoter	NN	O	VP	O	O
8	may	may	MD	O	VP	O	O
9	be	be	VB	O	VP	O	O
10	regulated	regulate	VBN	O	VP	O	O
11	by	by	IN	O	VP	O	O
12	the	the	DT	O	VP	O	O
13	degree	degree	NN	O	VP	O	O
14	of	of	IN	O	VP	O	O
15	squamous	squamous	JJ	O	VP	O	O
16	differentiation	differentiation	NN	O	VP	L	L
17	.	.	.	O	VP	O	O

**Table 2.** Learning Instance after Iteration 1 for sentence 3, hedge cue ‘suggests’

voice use, hedge scopes matched the syntactic scope of ancestors in the tree other than the parent. To cope with this, we added the same syntactic information for hedge cue grandparents and great-grandparents in the syntax tree, and their part-of-speech tags, exactly the same way we did in the previous section. We fed the classifier with all of them in the form of knowledge rules, and let it select which scope to use, depending on the remaining attributes. Performance, again, improve clearly, yielding a 0.740 F-score, eight points above the baseline, and four points above the previous iteration.

#### 4.4 Iteration 3: Adjusting Ancestor Scopes

Until this iteration we had assumed that syntactic scopes matched hedge scopes, i.e. that we could find an ancestor of the hedge cue in the syntax tree whose scope matched the scope of the hedge cue. Studying classification errors, we found that this assumption was not always true. In this iteration we aimed to see whether this was a general problem, and whether we could adjust constituent scopes to achieve concordance.

We first studied, for every hedge cue in the training corpus, if its hedge scope matched the syntactic scope of one of its ancestors, resulting from the sentence parsing. We found that for about 80% of the hedge cues this was actually true. To improve this matching, we studied the cases of misalignment where they accounted for a greater proportion and greater number of instances. We found two main causes for misalignments: the scope of the hedge cue included several non-nested scopes, not allowing a characterization of the hedge scope in terms of syntactic scopes, or the hedge scope matched just a portion of an ancestor scope, excluding some subconstituents. In both cases, scopes should be modified (adding or eliminating subcomponents) to match hedge scopes. Consider, for example, the following sentence:

*Example 4.* Our results lend further support to the {hypothesis that inflammatory and immune responses of monocytes/macrophages may be modulated at

the molecular level by signals originating from tissue structural cells such as fibroblasts}.

Its syntactic analysis tree (partially depicted in Figure 2), shows that the hedge cue scope (shown underlined) includes the parent noun phrase of the hedge cue (excluding the initial determiner) and the clause to the right of the grand-parent constituent. We can observe that this mismatch was probably originated by an incorrect parsing decision. The clause starting with ‘that inflammatory...’ could be considered within the syntactic scope of the NP clause headed by the noun hedge cue ‘hypothesis’. Several other, similar errors appeared in the held out corpus.

Recall that to obtain sentence constituents we used an external parser: therefore, it was not possible (in our working scenario) to correct parsing errors such as the one shown. However, analyzing the relation between hedge scopes and syntactic constituents in the training corpus it was possible for us to derive a series of rules to ‘correct’ the extracted features and adjust the syntactic scopes based mainly on lexical information. This procedure did not aim to derive a correct parse, but only to produce better features for the task, in certain clearly identified cases. Similar comments could be made to those cases where the syntax constituent was correctly derived, but, due to annotation idiosyncrasies, certain parts should not be included in the hedge scope.

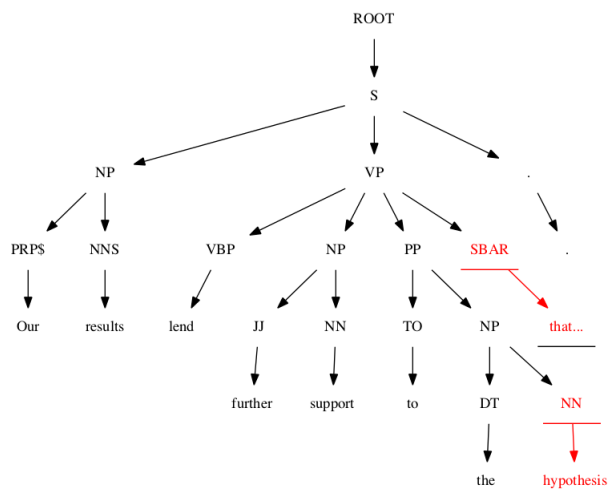


Fig. 2. Hedge scope for sentence 4

After looking at several of these mismatches, we modified our definition of ‘constituent scope’, incorporating several rules to cope with specific, ‘pathological’ cases. Once again, performance improved on the held out corpus, but this time only by 1.5 points, giving an F-measure of 0.756.



#### 4.5 Iteration 4: Handling Misclassified examples

Another source of errors beside wrong scope selection and mismatches between hedge and syntactic scopes, were the cases where the classifier failed to classify one sentence token as the first element of the scope and one token as the last one. In these cases, we could be sure that this evaluation instances would be misclassified, because they were ‘badly-formed’, since they did not meet the very definition of scope. After the last iteration, more than half of classification errors corresponded to these cases.

As we have previously mentioned, these cases were handled using post-processing rules, based mainly in positional and lexical information. Since we had available the sentence syntax structure, we decided to modify these rules and base them on syntax, probably reflecting better the corpus annotation guidelines. We ended using only three post-processing rules:

- If the hedge cue is a conjugated verb (except when in passive voice or in the case of raising verbs such as ‘seem’), use the next verb phrase up in the syntax tree that includes it.
- If the hedge cue is ‘or’, ‘neither’ or ‘either’, use the first noun phrase that includes it.
- In every other case, use the first clause that includes the hedge cue.

After modifying post-processing rules, performance dramatically improved by more than ten percentage points, obtaining a 0.860 F-measure value.

#### 4.6 Iteration 5: Post-processing Rules

After we had studied the three types of errors identified in section 4.3 and modified the classifier attributes and post-processing rules, improving classification, we had only 49 errors left. We studied these errors, and identified several patterns where the classifier did not manage to predict the correct scopes. We found that in most cases these errors probably corresponded to situations where, despite having enough attributes, the absence of enough training data prevented the classifier to infer them. To try to solve this problem, we took a rule-based approach: we ignored the classifier predictions and deterministically assigned the scope limits. We had to be careful about being very precise in the determination of the situations these rules fired, the avoid introducing false positives. After applying these rules, and fine tuning parameters, performance improved to its best value on the held out corpus with 0.885 in terms of F-measure.

#### 4.7 Results on the Evaluation Corpus

To estimate the performance of our classifier, we rebuilt the classifiers in each one of the methods iteration, this time training on the whole training corpus (also including sentences in the held out corpus). We do not only wish to estimate the prediction ability of our best classifier on future data, but also to

evaluate if the application of the methodology actually produced a performance improvement after each iteration, when faced with unseen data. This is of particular importance for our method: since improvement was based on errors on a held out corpus, we risked overfitting our built classifiers to this corpus. Table 3 summarizes the results<sup>6</sup>.

Classifier	F-score
Baseline	0.737 (0.664)
It. #1 (Adding syntax info)	0.749 (0.705)
It. #2 (Adding ancestors)	0.802 (0.740)
It. #3 (Adjusting ancestors scopes)	0.800 (0.756)
It. #4 (Handling misclassified examples)	<b>0.852</b> (0.860)
It. #5 (Post-processing rules)	0.831 ( <b>0.885</b> )

**Table 3.** Scope detection: results on evaluation data. Number in parenthesis show results on the held out corpus.

It can be observed that improvements in the held out corpus hold for the evaluation corpus, except for iteration #3 and #5. In both cases, overfitting occurred. It seems that specific rules used to adjust scopes or even final results were too tailored to solve errors in the held out corpus, and actually harmful for the overall performance on unseen data. On the other hand, general decisions, such as the addition of new syntax attributes or a general modification of post-processing rules clearly improved performance. At the end of the day, the best classifier improved performance by more than twelve points in term of F-measure, compared to the baseline classifier.

## 5 Final Remarks

In this article we presented a series of experiment on the detection of hedge scopes, and showed how we improved classifier performance incorporating attributes resulting from syntactic analysis. All the results presented suggest that the methodology proposed successfully identified syntax information as an important predictor of hedge scopes, clearly improving baseline results for scope detection.

However, there were still several instances that could not be correctly classified, especially those that did not correspond with the scope of any ancestor of the hedge cue. We found that the classification method used was highly successful in finding the correlation between syntactic scopes and hedge scopes, when

<sup>6</sup> Note that these results are not comparable with state-of-the art numbers, since most of them are calculated on the evaluation corpus of the CoNLL 2010 Shared Task, which, for example, includes full articles. Actually, we evaluated our methods against the CoNLL corpus, obtaining competitive results. See Moncecchi et al. (2013) for details.

that was possible. In almost every case the correct ancestor in the tree was selected by the classifier to be used as the hedge scope, independently of the hedge cue part-of-speech tag. Adjusting the original syntactic scope of the hedge cue ancestor to exclude or add certain spans (such as certain prepositional phrases or clauses at the end or at the beginning of the scope) seemed useful to improve classification, but produced no improvement on the evaluation data. We think that the absence of clear annotation rules for the selected corpus, indicating when they should be included or excluded acted as a source of ambiguity, and we found several corpus instances where the reasons for pruning or not pruning the hedge scope were not clear. The rules for scope pruning that we developed were based purely on data observation, causing an overfit to the training corpus.

Another source of hedge scope detection errors was syntax parsing. Several of the remaining errors on the held out corpus corresponded to cases such as the one presented in section 4.4, where syntax constituents were incorrectly identified by the parser, yielding wrong syntactic scopes and consequently wrong hedge scopes. Improving parsing results would almost surely imply improving hedge scope identification.

In future research, we plan to apply this methodology to similar phenomena involving cues and scope, such as negation, event recognition, factuality or modality identification. Previous work on those topics suggests that they present several similarities and relation with syntax, with respect to the here addressed phenomenon of hedging.

## References

- Buchholz, S., Marsi, E.: Conll-x shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning. pp. 149–164. Association for Computational Linguistics (2006)
- Farkas, R., Vincze, V., Móra, G., Csirik, J., Szarvas, G.: 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. pp. 1–12. Association for Computational Linguistics, Uppsala, Sweden (July 2010)
- Hyland, K.: The author in the text: Hedging scientific writing. *Hongkong Papers in Linguistics and Language Teaching* 18, 33–42 (1995)
- Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. pp. 423–430. Association for Computational Linguistics, Morristown, NJ, USA (2003)
- Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML-01. pp. 282–289 (2001)
- Lakoff, G.: Hedges: A study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic* 2(4), 458–508 (October 1973)
- Medlock, B., Briscoe, T.: Weakly supervised learning for hedge classification in scientific literature. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (2007)
- Moncecchi, G.: Recognizing Speculative Language in Research Texts., Phd. Thesis, Universidad de la República, Montevideo, Uruguay - Université Paris Ouest Nanterre La Défense (2013)

- Moncecchi, G., Minel, J.L., Wonsever, D.: Enriching the bioscope corpus with lexical and syntactic information. In: *Workshop in Natural Language Processing and Web-based Technologies 2010*. pp. 137–146 (Nov 2010)
- Moncecchi, G., Minel, J.L., Wonsever, D.: Improving speculative language detection using linguistic knowledge. In: *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*. p. 37–46. Association for Computational Linguistics, Association for Computational Linguistics, Jeju, Republic of Korea (2012)
- Morante, R., Daelemans, W.: Learning the scope of hedge cues in biomedical texts. In: *Proceedings of the BioNLP 2009 Workshop*. pp. 28–36. Association for Computational Linguistics, Boulder, Colorado (June 2009)
- Morante, R., Van Asch, V., Daelemans, W.: Memory-based resolution of in-sentence scopes of hedge cues. In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. pp. 40–47. Association for Computational Linguistics, Uppsala, Sweden (July 2010)
- Øvrelid, L., Velldal, E., Oepen, S.: Syntactic scope resolution in uncertainty analysis. In: *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*. pp. 1379–1387. Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
- Özgür, A., Radev, D.R.: Detecting speculations and their scopes in scientific text. In: *EMNLP*. Singapore (2009)
- Tsuruoka, Y., Tateishi, Y., Kim, J.D., Ohta, T., McNaught, J., Ananiadou, S., Tsujii, J.: Developing a robust Part-of-Speech tagger for biomedical text. In: Bozanis, P., Houstis, E.N. (eds.) *Advances in Informatics*, vol. 3746, chap. 36, pp. 382–392. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
- Velldal, E., Øvrelid, L., Oepen, S.: Resolving speculation: Maxent cue classification and dependency-based scope rules. In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. pp. 48–55. Association for Computational Linguistics, Uppsala, Sweden (July 2010)
- Velldal, E., Øvrelid, L., Read, J., Oepen, S.: Speculation and negation: Rules, rankers, and the role of syntax. *Computational Linguistics* pp. 1–64 (Feb 2012)
- Vincze, V., Szarvas, G., Farkas, R., Mora, G., Csirik, J.: The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics* 9(Suppl 11), S9+ (2008)