



HAL
open science

Improving speculative language recognition incorporating expert knowledge

Guillermo Moncecchi, Jean-Luc Minel, Dina Wonsever

► **To cite this version:**

Guillermo Moncecchi, Jean-Luc Minel, Dina Wonsever. Improving speculative language recognition incorporating expert knowledge. Roser Morante, Caroline Sporleder. ExProM 2012, Jul 2012, Jeju, South Korea. Association for Computational Linguistics (ACL), pp.37-46, 2012. halshs-00721590

HAL Id: halshs-00721590

<https://shs.hal.science/halshs-00721590v1>

Submitted on 28 Jul 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improving Speculative Language Detection using Linguistic Knowledge

Guillermo Moncecchi Facultad de Ingeniería Universidad de la República Montevideo, Uruguay

Jean-Luc Minel Laboratoire MoDyCo Université Paris Ouest Nanterre La Défense, France

Dina Wonsever Facultad de Ingeniería Universidad de la República Montevideo, Uruguay

Abstract

In this paper we present an iterative methodology to improve classifier performance by incorporating linguistic knowledge, and propose a way to incorporate domain rules into the learning process. We applied the methodology to the tasks of hedge cue recognition and scope detection and obtained competitive results on a publicly available corpus.

1 Introduction

A common task in Natural Language Processing (NLP) is to extract or infer factual information from textual data. In the field of natural sciences this task turns out to be of particular importance, because science aims to discover or describe facts from the world around us. Extracting these facts from the huge and constantly growing body of research articles in areas such as, for example, molecular biology, becomes increasingly necessary, and has been the subject of intense research in the last decade (Ananiadou et al., 2006). The fields of information extraction and text mining have paid particular attention to this issue, seeking to automatically populate structured databases with data extracted or inferred from text. In both cases, the problem of speculative language detection is a challenging one, because it may correspond to a subjective attitude of the writer towards the truth value of certain facts, and that information should not be lost when the fact is extracted or inferred.

When researchers express facts and relations in their research articles, they often use speculative language to convey their attitude to the truth of what

is said. Hedging, a term first introduced by Lakoff (1973) to describe ‘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. Hyland (1996a) reports one hedge in every 50 words of a corpus of research articles; Light et al. (2004) mention that 11% of the sentences in MEDLINE contain speculative language. Vincze et al. (2008) report that 18% of the sentences in the scientific abstracts section of the Bioscope corpus correspond to speculations.

Early work on speculative language detection tried to classify a sentence either as speculative or non-speculative (see, for example, Medlock and Briscoe (2007)). This approach does not take into account the fact that hedging usually affects propositions or claims (Hyland, 1995) and that sentences often include more than one of them. When the Bioscope corpus (Vincze et al., 2008) was developed the notions of hedge cue (corresponding to what was previously called just ‘hedges’ in the literature) and scope (the propositions affected by the hedge cues) were introduced. In this context, speculative language recognition can be seen as a two-phase process: first, the existence of a hedge cue in a sentence is detected, and second, the scope of the induced hedge is determined. This approach was first used by Morante et al. (2008) and subsequently in many of the studies presented in the CoNLL-2010 Conference Shared Task (Farkas et al., 2010a), and is the one used in this paper.

For example, the sentence

- (1) This finding {suggests suggests that the BZLF1

promoter {_{may} may be regulated by the degree of squamous differentiation }_{may} }_{suggests}.

contains the word ‘may’ that acts as a hedge cue (i.e. attenuating the affirmation); this hedge only affects the propositions included in the subordinate clause that contains it.

Each of these phases can be modelled (albeit with some differences, described in the following sections) as a sequential classification task, using a similar approach to that commonly used for named entity recognition or semantic labelling: every word in the sentence is assigned a class, identifying spans of text (as, for example, scopes) with, for example, a special class for the first and last element of the span. Correctly learning these classes is the computational task to be solved.

In this paper we present a methodology and machine learning system implementing it that, based on previous work on speculation detection, studies how to improve recognition by analysing learning errors and incorporating advice from domain experts in order to solve the errors without hurting overall performance. The methodology proposes the use of domain knowledge rules that suggest a class for an instance, and shows how to incorporate them into the learning process. In our particular task domain knowledge is linguistic knowledge, as hedging and scopes issues are general linguistic devices. In this paper we are going both terms interchangeably.

The paper is organized as follows. In Section 2 we review previous theoretical work on speculative language and the main computational approaches to the task of detecting speculative sentences. Section 3 briefly describes the corpus used for training and evaluation. In Section 4 we present the specific computational task to which our methodology was applied. In Section 5 we present the learning methodology we propose to use, and describe the system we implemented, including lexical, syntactic and semantic attributes we experimented with. We present and discuss the results obtained in Section 6. Finally, in Section 7 we analyse the approach presented here and discuss its advantages and problems, suggesting future lines of research.

2 Related work

The grammatical phenomenon of modality, defined as ‘a category of linguistic meaning having to do with the expression of possibility and necessity’ (von Stechow, 2006) has been extensively studied in the linguistic literature. Modality can be expressed using different linguistic devices: in English, for example, modal auxiliaries (such as ‘could’ or ‘must’), adverbs (‘perhaps’), adjectives (‘possible’), or other lexical verbs (‘suggest’, ‘indicate’), are used to express the different ways of modality. Other languages express modality in different forms, for example using the subjunctive mood. Palmer (2001) considers modality as the grammaticalization of speakers’ attitudes and opinions, and epistemic modality, in particular, applies to ‘any modal system that indicates the degree of commitment by the speaker to what he says’.

Although hedging is a concept that is closely related to epistemic modality, they are different: modality is a grammatical category, whereas hedging is a pragmatic position (Morante and Sporleder, 2012). This phenomenon has been theoretically studied in different domains and particularly in scientific writing (Hyland, 1995; Hyland, 1996b; Hyland, 1996a).

From a computational point of view, speculative language detection is an emerging area of research, and it is only in the last five years that a relatively large body of work has been produced. In the remainder of this section, we survey the main approaches to hedge recognition, particularly in English and in research discourse.

Medlock and Briscoe (2007) applied a weakly supervised learning algorithm to classify sentences as speculative or non-speculative, using a corpus they built and made publicly available. Morante and Daelemans (2009) not only tried to detect hedge cues but also to identify their scope, using a meta-learning approach based on three supervised learning methods. They achieved an F1 of 84.77 for hedge identification, and 78.54 for scope detection (using gold-standard hedge signals) in the Abstracts sections of the Bioscope corpus.

Task 2 of the CoNLL-2010 Conference Shared Task (Farkas et al., 2010b) proposed solving the problem of in-sentence hedge cue phrase identi-

fication and scope detection in two different domains (biological publications and Wikipedia articles), based on manually annotated corpora. The evaluation criterion was in terms of precision, recall and F-measure, accepting a scope as correctly classified if the hedge cue and scope boundaries were both correctly identified.

The best result on hedge cue identification (Tang et al., 2010) obtained an F-score of 81.3 using a supervised sequential learning algorithm to learn BIO classes from lexical and shallow parsing information, also including certain linguistic rules. For scope detection, Morante et al. (2010) obtained an F-score of 57.3, using also a sequence classification approach for detecting boundaries (tagged in FOL format, where the first token of the span is marked with an F, while the last one is marked with an L). The attributes used included lexical information, dependency parsing information, and some features based on the information in the parse tree.

The approximation of Velldal et al. (2010) for scope detection was somewhat different: they developed a set of handcrafted rules, based on dependency parsing and lexical features. With this approach, they achieved an F-score of 55.3, the third best for the task. Similarly, Kilicoglu and Bergler (2010) used a pure rule-based approach based on constituent parse trees in addition to syntactic dependency relations, and achieved the fourth best F-score for scope detection, and the highest precision of the whole task (62.5). In a recent paper, Velldal et al. (2012) reported a better F-score of 59.4 on the same corpus for scope detection using a hybrid approach that combined a set of rules on syntactic features and n-gram features of surface forms and lexical information and a machine learning system that selected subtrees in constituent structures.

3 Corpus

The system presented in this paper uses the Bioscope corpus (Vincze et al., 2008) as a learning source and for evaluation purposes. The Bioscope corpus is a freely available corpus of medical free texts, biological full papers and biological abstracts, annotated at a token level with negative and speculative keywords, and at sentence level with their linguistic scope.

	Clinical	Full	Abstract
#Documents	954	9	1273
#Sentences	6383	2670	11871
%Hedge Sentences	13.4	19.4	17.7
#Hedge cues	1189	714	2769

Table 1: Bioscope corpus statistics about hedging

Table 1, extracted from Vincze et al. (2008), gives some statistics related to hedge cues and sentences for the three sub corpora included in Bioscope.

For the present study, we use only the Abstract sub corpus for training and evaluation. We randomly separated 20% of the corpus, leaving it for evaluation purposes. We further sub-divided the remaining training corpus, separating another 20% that was used as a held out corpus. All the models presented here were trained on the resulting training corpus and their performance evaluated on the held out corpus. The final results were computed on the previously unseen evaluation corpus.

4 Task description

From a computational point of view, both hedge cue identification and scope detection can be seen as a sequence classification problem: given a sentence, classify each token as part of a hedge cue (or scope) or not. In almost every classification problem, two main approaches can be taken (although many variations and combinations exist in the literature): build the classifier as a set of handcrafted rules, which, from certain attributes of the instances, decide which category it belongs to, or learn the classifier from previously annotated examples, in a supervised learning approach.

The rules approach is particularly suitable when domain experts are available to write the rules, and when features directly represent linguistic information (for example, POS-tags) or other types of domain information. It is usually a time-consuming task, but it probably grasps the subtleties of the linguistic phenomena studied better, making it possible to take them into account when building the classifier. The supervised learning approach needs tagged data; in recent years the availability of tagged text

has grown, and this type of method has become the state-of-the-art solution for many NLP problems.

In our particular problem, we have both tagged data and expert knowledge (represented by the body of work on modality and hedging), so it seems reasonable to see how we can combine the two methods to achieve better classification performance.

4.1 Identifying hedge cues

The best results so far for this task used a token classification approach or sequential labelling techniques, as Farkas et al. (2010b) note. In both cases, every token in the sentence is assigned a class label indicating whether or not that word is acting as a hedge cue. To allow for multiple-token hedge cues, we identify the first token of the span with the class **B** and every other token in the span with **I**, keeping the **O** class for every token not included in the span, as the following example shows:

(2) The/O findings/O indicate/B that/I MNDA/O expression/O is/O ... [401.8]

After token labelling, hedge cue identification can be seen as the problem of assigning the correct class to each token of an unlabelled sentence. Hedge cue identification is a sequential classification task: we want to assign classes to an entire ordered sequence of tokens and try to maximize the probability of assigning the correct classes to every token in the sequence, considering the sequence as a whole, not just as a set of isolated tokens.

4.2 Determining the scope of hedge cues

The second sub-task involves marking the part of the sentence affected by the previously identified hedge cue. Scopes are also spans of text (typically longer than multi-word hedge cues), so we could use the same reduction to a token classification task. Being longer, FOL classes are usually used for classification, identifying the first token of the scope as **F**, the last token as **L** and any other token in the sentence as **O**. Scope detection poses an additional problem: hedge cues cannot be nested, but scopes (as we have already seen) usually are. In example 1, the scope of ‘may’ is nested within the scope of ‘suggests’. To overcome this, Morante and Daelemans (2009) propose to generate a different learning example for each cue in the sen-

tence. In this setting, each example becomes a pair labelled sentence, hedge cue position. So, for example 1, the scope learning instances would be:

(3) hThis/O finding/O suggests/F that/O the/O BZLF1/O promoter/O may/O be/O regulated/O by/O the/O degree/O of/O squamous/O differentiation/L./O, 3i

(4) hThis/O finding/O suggests/O that/O the/F BZLF1/O promoter/O may/O be/O regulated/O by/O the/O degree/O of/O squamous/O differentiation/L./O, 8i

Learning on these instances, and using a similar approach to the one used in the previous task, we should be able to identify scopes for previously unseen examples. Of course, the two tasks are not independent: the success of the second one depends on the success of the first. Accordingly, evaluation of the second task can be done using gold standard hedge cues or with the hedge cues learned in the first task.

5 Methodology and System Description

To approach both sequential learning tasks, we follow a learning methodology (depicted in Figure 1), that starts with an initial guess of attributes for supervised learning and a learning method, and tries to improve its performance by incorporating domain knowledge. We consider that expressing this knowledge through rules (instead of learning features) is a better way for a domain expert to suggest new useful information or to generalize certain relations between attributes and classification results when the learning method cannot achieve this because of insufficient training data. These rules, of course, have to be converted to attributes to incorporate them into the learning process. These attributes are what we call knowledge rules and their generation will be described in the Analysis section.

5.1 Preprocessing

Before learning, we propose to add every possible item of external information to the corpus so as to integrate different sources of knowledge (either the result of external analysis or in the form of semantic resources). After this step, all the information is consolidated into a single structure, facilitating

subsequent analysis. In our case, we incorporate POS-tagging information, resulting from the application of the GENIA tagger (Tsuruoka et al., 2005), and deep syntax information obtained with the application of the Stanford Parser (Klein and Manning, 2003), leading to a syntax-oriented representation of the training data. For a detailed description of the enriching process, the reader is referred to Moncechi et al. (2010).

5.2 Initial Classifier

The first step for improving performance is, of course, to select an initial set of learning features, and learn from training data to obtain the first classifier, in a traditional supervised learning scenario. The sequential classification method will depend on the addressed task. After learning, the classifier is applied on the held out corpus to evaluate its performance (usually in terms of Precision, Recall and F1-measure), yielding performance results and a list of errors for analysis. This information is the source for subsequent linguistic analysis. As such, it seems important to provide ways to easily analyse instance attributes and learning errors. For our tasks, we have developed visualization tools to inspect the tree representation of the corpus data, the learning attributes, and the original and predicted classes.

5.3 Analysis

From the classifier results on the held-out corpus, an analysis phase starts, which tries to incorporate linguistic knowledge to improve performance.

One typical form of introducing new information is through learning features: for example, we can add a new attribute indicating if the current instance (in our case, a sentence token) belongs to a list of common hedge cues.

However, linguistic or domain knowledge can also naturally be stated as rules that suggest the class or list of classes that should be assigned to instances, based on certain conditions on features, linguistic knowledge or data observation. For example, based on corpus annotation guidelines, a rule could state that the scope of a verb hedge cue should be the verb phrase that includes the cue, as in the expression

(5) This finding {_{suggests} suggests that the BZLF1 promoter may be regulated by the degree of

squamous differentiation }_{suggests}.

We assume that these rules take the form ‘if a condition C holds then classify instance X with class Y’. In the previous example, assuming a FOL format for scope identification, the token ‘suggest’ should be assigned class F and the token ‘differentiation’ should be assigned class L, assigning class O to every other token in the sentence.

The general problem with these rules is that as we do not know in fact if they always apply, we do not want to directly modify the classification results, but to incorporate them as attributes for the learning task. To do this, we propose to use a similar approach to the one used by Rosá (2011), i.e. to incorporate these rules as a new attribute, valued with the class predictions of the rule, trying to ‘help’ the classifier to detect those cases where the rule should fire, without ignoring the remaining attributes. In the previous example, this attribute would be (when the rule condition holds) valued F or L if the token corresponds to the first or last word of the enclosing verb phrase, respectively. We have called these attributes knowledge rules to reflect the fact that they suggest a classification result based on domain knowledge.

This configuration allows us to incorporate heuristic rules without caring too much about their potential precision or recall ability: we expect the classification method to do this for us, detecting correlations between the rule result (and the rest of the attributes) and the predicted class.

There are some cases where we do actually want to overwrite classifier results: this is the case when we know the classifier has made an error, because the results are not well-formed. For example, we have included a rule that modifies the assigned classes when the classifier has not exactly found one F token and one L token, as we know for sure that something has gone wrong. In this case, we decided to assign the scope based on a series of postprocessing rules: for example, assign the scope of the enclosing clause in the syntax tree as hedge scope, in the case of verb hedge cues.

For sequential classification tasks, there is an additional issue: sometimes the knowledge rule indicates the beginning of the sequence, and its end can be determined using the remaining attributes. For example, suppose the classifier suggests the class

Hedge	PPOS	GPPOS	Lemma	PScope	GPScope	Scope
O	VP	S	This	O	O	O
O O	VP	S S	finding	O O	O O	O O
O O	VP	S S	suggest	O O	O F	O F
O O	VP	S S	that	O O	O O	O O
B O	VP	S S	the BZLF1	F O	O O	O O
O O	VP	S S	promoter	O O	O O	O O
O O	VP	S S	may	O O	O O	O O
O O	VP	S S	be	O O	O O	O O
O O	VP	S S	regulate	L O	L O	O O
VP			by			
VP			the			
VP			degree of			
VP			squamous			
VP			differentiation			
VP			.			

Table 2: Evaluation instance where the scope ending could not be identified

scope in the learning instance shown in table 2 (using as attributes the scopes of the parent and grandparent constituents for the hedge cue in the syntax tree). If we could associate the F class suggested by the classifier with the grand parent scope rule, we would not be concerned about the prediction for the last token, because we would know it would always correspond to the last token of the grand parent clause. To achieve this, we modified the class we want to learn, introducing a new class, say X, instead of F, to indicate that, in those cases, the L token must not be learned, but calculated in the postprocessing step, in terms of other attributes' values (in this example, using the hedge cue grandparent constituent limits). This change also affects the classes of training data instances (in the example, every training instance where the scope coincides with the grand parent scope attribute will have its F-classified token class changed to X).

In the previous example, if the classifier assigns class X to the 'the' token, the postprocessing step will change the class assigned to the 'differentiation' token to L, no matter which class the classifier had predicted, changing also the X class to the original F, yielding a correctly identified scope.

After adding the new attributes and changing the relevant class values in the training set, the process starts over again. If performance on the held out corpus improves, these attributes are added to the best configuration so far, and used as the starting point for a new analysis. When no further improvement can be achieved, the process ends, yielding the best

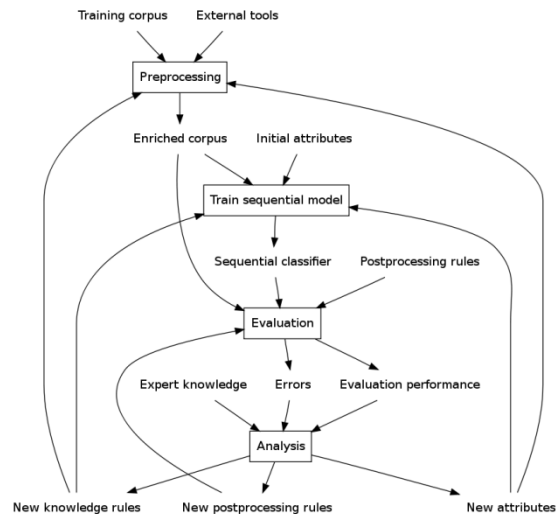


Figure 1: Methodology overview

classifier as a result.

We applied the proposed methodology to the tasks of hedge cue detection and scope resolution. We were mainly interested in evaluating whether systematically applying the methodology would indeed improve classifier performance. The following sections show how we tackled each task, and how we managed to incorporate expert knowledge and improve classification.

5.4 Hedge Cue Identification

To identify hedge cues we started with a sequential classifier based on Conditional Random Fields (Lafferty et al., 2001), the state-of-the-art classification method used for sequence supervised learning in many NLP tasks. The baseline configuration we started with included a size-2 window of surface forms to the left and right of the current token, pairs and triples of previous/current surface forms. This led to a highly precise classifier (an F-measure of 95.5 on the held out corpus). After a grid search on different configurations of surface forms, lemmas and POS tags, we found (somewhat surprisingly) that the best precision/recall tradeoff was obtained just using a window of size 2 of unigrams of surface forms, lemmas and tokens with a slightly worse precision than the baseline classifier, but compen-

Configuration	P	R	F1
Baseline	95.5	74.0	83.4
Conf1	94.7	80.3	86.9
Conf2	91.3	84.0	87.5

Table 3: Classification performance on the held out corpus for hedge cue detection. Conf1 corresponds to windows of Word, Lemma and POS attributes and Conf2 incorporates hedge cue candidates and cooccurring words

sated by an improvement of about six points in recall, achieving an F-score of 86.9.

In the analysis step of the methodology we found that most errors came from False Negatives, i.e. words incorrectly not marked as hedges. We also found that those words actually occurred in the training corpus as hedge cues, so we decided to add new rule attributes indicating membership to certain semantic classes. After checking the literature, we added three attributes:

- Hyland words membership: this feature was set to Y if the word was part of the list of words identified by Hyland (2005)
- Hedge cue candidates: this feature was set to Y if the word appeared as a hedge cue in the training corpus
- Words co-occurring with hedge cue candidates: this feature was set to Y if the word cooccured with a hedge cue candidate in the training corpus. This feature is based on the observation that 43% of the hedges in a corpus of scientific articles occur in the same sentence as at least another device (Hyland, 1995).

After adding these attributes and tuning the window sizes, performance improved to an F-score of 87.5 in the held-out corpus

5.5 Scope identification

To learn scope boundaries, we started with a similar configuration of a CRF classifier, using a window of size 2 of surface forms, lemmas and POS-tags, and the hedge cue identification attribute (either obtained from the training corpus when using gold standard

hedge cues or learned in the previous step), achieving a performance of 63.7 in terms of F-measure. When we incorporated information in the form of a knowledge rule that suggested the scope of the constituent of the parsing tree headed by the parent node of the first word of the hedge cue, and an attribute containing the parent POS-tag, performance rapidly improved about two points measured in terms of F-score.

After several iterations, and analyzing classification errors, we included several knowledge rules, attributes and postprocessing rules that dramatically improved performance on the held-out corpus:

- We included attributes for the scope of the next three ancestors of the first word of the hedge cue in the parsing tree, and their respective POS-tags, in a similar way as with the parent. We also included a trigram with the ancestors POS from the word upward in the tree.
- For parent and grandparent scopes, we incorporated X and Y classes instead of F, and modified postprocessing to use the last token of the corresponding scope when one of these classes was learned.
- We modified the ancestors scopes to reflect some corpus annotation guidelines or other criteria induced after data examination. For example, we decided not to include adverbial phrases or prepositional phrases at the beginning of scopes, when they corresponded to a clause, as in

(6) In addition, {unwanted and potentially hazardous specificities may be elicited... }

- We added postprocessing rules to cope with cases where (probably due to insufficient training data), the classifier misclassified certain instances. For example, we forced classification to use the next enclosing clause (instead of verb phrase), when the hedge cue was a verb conjugated in passive voice, as in

(7) {GATA3 , a member of the GATA family that is abundantly expressed in the T-lymphocyte lineage , is thought to participate in ... }.

Configuration	Gold-P	P	R	F1
Baseline	66.4	68.6	59.6	63.8
Conf1	68.7	71.3	61.8	66.2
Conf2	73.3	75.6	65.4	70.1
Conf3	80.9	82.1	71.3	76.3
Conf4	88.2	82.0	76.3	79.1

Table 4: Classification performance on the held out corpus. The baseline used a window of Word, Lemma, POS attributes and hedge cue tag; Conf1 included parent scopes, Conf2 added grandparents information; Conf3 added postprocessing rules. Finally, Conf4 used adjusted scopes and incorporated new postprocessing rules

- We excluded references at the end of sentences from all the calculated scopes.
- We forced classification to the next S,VP or NP ancestor constituent in the syntax tree (depending on the hedge cue POS), when full scopes could not be determined by the statistical classifier (missing either L or F, or learning more than one of them in the same sentence).

Table 4 summarizes the results of scope identification in the held out corpus. The first results were obtained using gold-standard hedge cues, while the second ones used the hedge cues learned in the previous step (for hedge cue identification, we used the best configuration we found). In the gold-standard results, Precision, Recall and the F-measure are the same because every False Positive (incorrectly marked scope) implied a False Negative (the missed right scope).

6 Evaluation

To determine classifier performance, we evaluated the classifiers found after improvement on the evaluation corpus. We also evaluated the less efficient classifiers to see whether applying the iterative improvement had overfitted the classifier to the corpus. To evaluate scope detection, we used the best configuration found in the evaluation corpus for hedge cue identification. Tables 5 and 6 show the results for the hedge cue recognition and scope resolution, respectively. In both tasks, classifier performance

Configuration	P	R	F1
Baseline	97.9	78.0	86.8
Conf1	95.9	84.9	90.1
Conf2	94.1	88.6	91.3

Table 5: Classification performance on the evaluation corpus for hedge cue detection

Configuration	Gold-P	P	R	F1
Baseline	74.0	71.9	68.1	70.0
Conf1	76.5	74.4	70.2	72.3
Conf2	80.0	77.2	72.9	75.0
Conf3	83.1	80.0	75.2	77.3
Conf4	84.7	80.1	75.8	77.9

Table 6: Classification performance on the evaluation corpus for scope detection

improved in a similar way to the results obtained on the held out corpus.

Finally, to compare our results with state-of-the-art methods (even though that was not the main objective of the study), we used the corpus of de CoNLL 2010 Shared Task to train and evaluate our classifiers, using the best configurations found in the evaluation corpus, and obtained competitive results in both subtasks of Task 2. Our classifier for hedge cue detection achieved an F-measure of 79.9, better than the third position in the Shared Task for hedge identification. Scope detection results (using learned hedge cues) achieved an F-measure of 54.7, performing better than the fifth result in the corresponding task, and five points below the best results obtained so far in the corpus (Velldal et al.,

	Hedge cue identification	Scope detection
Best results	81.7/81.0/81.3	59.6/55.2/57.3
Our results	83.2/76.8/79.9	56.7/52.8/54.7

Table 7: Classification performance compared with best results in CoNLL Shared Task. Figures represent Precision/Recall/F1-measure

2012). Table 7 summarizes these results in terms of Precision/Recall/F1-measure.

7 Conclusions and Future Research

In this paper we have presented an iterative methodology to improve classifier performance by incorporating linguistic knowledge, and proposed a way to incorporate domain rules to the learning process. We applied the methodology to the task of hedge cue recognition and scope finding, improving performance by incorporating information of training corpus occurrences and co-occurrences for the first task, and syntax constituents information for the second. In both tasks, results were competitive with the best results obtained so far on a publicly available corpus. This methodology could be easily used for other sequential (or even traditional) classification tasks.

Two directions are planned for future research: first, to improve the classifier results by incorporating more knowledge rules such as those described by Veldal et al. (2012) or semantic resources, specially for the scope detection task. Second, to improve the methodology, for example by adding some way to select the most common errors in the held out corpus and write rules based on their examination.

References

- S. Ananiadou, D. Kell, and J. Tsuj. 2006. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12):571–579, December.
- Richard Farkas, Veronika Vincze, Gyorgy Mora, Janos Csirik, and Gyo'rgy Szarvas. 2010a. 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*, pages 1–12, Uppsala, Sweden, July. Association for Computational Linguistics.
- Richárd Farkas, Veronika Vincze, György Szarvas, György Móra, and János Csirik, editors. 2010b. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Uppsala, Sweden, July.
- Ken Hyland. 1995. The author in the text: Hedging scientific writing. *Hongkong Papers in Linguistics and Language Teaching*, 18:33–42.
- Ken Hyland. 1996a. Talking to the academy: Forms of hedging in science research articles. *Written Communication*, 13(2):251–281.
- Ken Hyland. 1996b. Writing without conviction? Hedging in science research articles. *Applied Linguistics*, 17(4):433–454, December.
- Ken Hyland. 2005. *Metadiscourse: Exploring Interaction in Writing*. Continuum Discourse. Continuum.
- Halil Kilicoglu and Sabine Bergler. 2010. A high-precision approach to detecting hedges and their scopes. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 70–77, Uppsala, Sweden, July. Association for Computational Linguistics.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.
- John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*, pages 282–289.
- George Lakoff. 1973. Hedges: A study in meaning criteria and the logic of fuzzy concepts. *Journal of Philosophical Logic*, 2(4):458–508, October.
- Marc Light, Xin Y. Qiu, and Padmini Srinivasan. 2004. The language of bioscience: Facts, speculations, and statements in between. In Lynette Hirschman and James Pustejovsky, editors, *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 17–24, Boston, Massachusetts, USA, May. Association for Computational Linguistics.
- Ben Medlock and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.
- Guillermo Moncecchi, Jean-Luc Minel, and Dina Wonsever. 2010. Enriching the bioscope corpus with lexical and syntactic information. In *Workshop in Natural Language Processing and Web-based Technologies 2010*, pages 137–146, November.
- Roser Morante and Walter Daelemans. 2009. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the BioNLP 2009 Workshop*, pages 28–36, Boulder, Colorado, June. Association for Computational Linguistics.
- Roser Morante and Caroline Sporleder. 2012. Modality and negation: An introduction to the special issue. *Computational Linguistics*, pages 1–72, February.
- Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the scope of negation in biomedical texts. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 715–724, Morristown, NJ, USA. Association for Computational Linguistics.

Roser Morante, Vincent Van Asch, and Walter Daelemans. 2010. Memory-based resolution of in-sentence scopes of hedge cues. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 40–47, Uppsala, Sweden, July. Association for Computational Linguistics.

R. F. Palmer. 2001. Mood and Modality. Cambridge Textbooks in Linguistics. Cambridge University Press, New York.

Aiala Rosá. 2011. Identificación de opiniones de diferentes fuentes en textos en español. Ph.D. thesis, Universidad de la República (Uruguay), Université Paris Ouest (France), September.

Buzhou Tang, Xiaolong Wang, Xuan Wang, Bo Yuan, and Shixi Fan. 2010. A cascade method for detecting hedges and their scope in natural language text. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 13–17, Uppsala, Sweden, July. Association for Computational Linguistics.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust Part-of-Speech tagger for biomedical text. In Panayiotis Bozanis and Elias N. Houstis, editors, Advances in Informatics, volume 3746, chapter 36, pages 382–392. Springer Berlin Heidelberg, Berlin, Heidelberg.

Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2010. Resolving speculation: Maxent cue classification and dependency-based scope rules. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 48–55, Uppsala, Sweden, July. Association for Computational Linguistics.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers, and the role of syntax. Computational Linguistics, pages 1–64, February.

Veronika Vincze, Gyorgy Szarvas, Richard Farkas, Gyorgy Mora, and Janos Csirik. 2008. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. BMC Bioinformatics, 9(Suppl 11):S9+.

Kail von Fintel, 2006. Modality and Language. MacMillan Reference USA

