Uncertainty Detection as Approximate Max-Margin Sequence Labelling

Oscar Täckström
SICS / Uppsala University
Kista / Uppsala, Sweden
oscar@sics.se

Sumithra Velupillai
DSV, Stockholm University
Kista, Sweden
sumithra@dsv.su.se

Martin Hassel
DSV, Stockholm University
Kista, Sweden
xmartin@dsv.su.se

Gunnar Eriksson
SICS
Kista, Sweden
guer@sics.se

Hercules Dalianis
DSV, Stockholm University
Kista, Sweden
hercules@dsv.su.se

Jussi Karlgren
SICS
Kista, Sweden
jussi@sics.se

Abstract

This paper reports experiments for the CoNLL-2010 shared task on learning to detect hedges and their scope in natural language text. We have addressed the experimental tasks as supervised linear maximum margin prediction problems. For sentence level hedge detection in the biological domain we use an L_1-regularised binary support vector machine, while for sentence level weasel detection in the Wikipedia domain, we use an L_2-regularised approach. We model the in-sentence uncertainty cue and scope detection task as an L_2-regularised approximate maximum margin sequence labelling problem, using the BIO-encoding. In addition to surface level features, we use a variety of linguistic features based on a functional dependency analysis. A greedy forward selection strategy is used in exploring the large set of potential features. Our official results for Task 1 for the biological domain are 85.2 F_1-score, for the Wikipedia set 55.4 F_1-score. For Task 2, our official results are 2.1 for the entire task with a score of 62.5 for cue detection. After resolving errors and final bugs, our final results are for Task 1, biological: 86.0, Wikipedia: 58.2; Task 2, scopes: 39.6 and cues: 78.5.

1 Introduction

This paper reports experiments to detect uncertainty in text. The experiments are part of the two shared tasks given by CoNLL-2010 (Farkas et al., 2010). The first task is to identify uncertain sentences; the second task is to detect the cue phrase which makes the sentence uncertain and to mark its scope or span in the sentence.

Uncertainty as a target category needs to be addressed with some care. Sentences, utterances, statements are not uncertain – their producer, the speaker or author, is. Statements may explicitly indicate this uncertainty, employing several different linguistic and textual mechanisms to encode the speaker's attitude with respect to the veracity of an utterance. The absence of such markers does not necessarily indicate certainty – the opposition between certain and uncertain is not clearly demarkable, but more of a dimensional measure. Uncertainty on the part of the speaker may be difficult to differentiate from a certain assessment of an uncertain situation, *It is unclear whether this specimen is an X or a Y* vs. *The difference between X and Y is unclear*.

In this task, the basis for identifying uncertainty in utterances is almost entirely lexical. *Hedges*, the main target of this experiment, are an established category in lexical grammar analyses - see e.g. Quirk et al. (1985), for examples of English language constructions. Most languages use various verbal markers or modifiers for indicating the speaker's beliefs in what is being said, most prototypically using conditional or optative verb forms, *Six Parisiens seraient morts*, or auxiliaries, *This mushroom may be edible*, but aspectual markers may also be recruited for this purpose, more indirectly, *I'm hoping you will help* vs. *I hope you will help*; *Do you want to see me now* vs. *Did you want to see me now*. Besides verbs, there are classes of terms that through their presence, typically in an adverbial role, in an utterance make explicit its tentativeness: *possibly, perhaps...* and more complex constructions *with some reservation*, especially such that explicitly mention the speaker and the speaker's beliefs or doubts, *I suspect that X*.

Weasels, the other target of this experiment, on the other hand, do not indicate uncertainty.

Proceedings of the Fourteenth Conference on Computational Natural Language Learning: Shared Task, pages 84–91,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

Weasels are employed when speakers attempt to convince the listener of something they most likely are certain of themselves, by anchoring the truthfulness of the utterance to some outside fact or authority (*Most linguists believe in the existence of an autonomous linguistic processing component*), but where the authority in question is so unspecific as not to be verifiable when scrutinised.

We address both CoNLL-2010 shared tasks (Farkas et al., 2010). The first, detecting uncertain information on a sentence level, we solve by using an L_1-regularised support vector machine with hinge loss for the biological domain, and an L_2-regularised maximum margin model for the Wikipedia domain. The second task, resolution of in-sentence scopes of hedge cues, we approach as an approximate L_2-regularized maximum margin structured prediction problem. Our official results for Task 1 for the biological domain are 85.2 F_1-score, for the Wikipedia set 55.4 F_1-score. For Task 2, our official results were 2.1 for the entire task with a score of 62.5 for cue detection. After resolving errors and unfortunate bugs, our final results are for Task 1, biological: 86.0, Wikipedia: 58.2; Task 2: 39.6 and 78.5 for cues.

2 Detecting Sentence Level Uncertainty

On the sentence level, word- and lemma-based features have been shown to be useful for uncertainty detection (see e.g. Light et al. (2004), Medlock and Briscoe (2007), Medlock (2008), and Szarvas (2008)). Medlock (2008) and Szarvas (2008) employ probabilistic, weakly supervised methods, where in the former, a stemmed single term and bigram representation achieved best results (0.82 BEP), and in the latter, a more complex n-gram feature selection procedure was applied using a Maximum Entropy classifier, achieving best results when adding reliable keywords from an external hedge keyword dictionary (0.85 BEP, 85.08 F_1-score on biomedical articles). More linguistically motivated features are used by Kilicoglu and Bergler (2008), such as negated "unhedging" verbs and nouns and *that* preceded by epistemic verbs and nouns. On the fruit-fly dataset (Medlock and Briscoe, 2007) they achieve 0.85 BEP, and on the BMC dataset (Szarvas, 2008) they achieve 0.82 BEP. Light et al. (2004) also found that most of the uncertain sentences appeared towards the end of the abstract, indicating that the position of an uncertain sentence might be a useful feature.

Ganter and Strube (2009) consider weasel tags in Wikipedia articles as hedge cues, and achieve results of 0.70 BEP using word- and distance based features on a test set automatically derived from Wikipedia, and 0.69 BEP on a manually annotated test set using syntactic patterns as features. These results suggest that syntactic features are useful for identifying weasels that ought to be tagged. However, evaluation is performed on balanced test sets, which gives a higher baseline.

2.1 Learning and Optimization Framework

A guiding principle in our approach to this shared task has been to focus on highly computationally efficient models, both in terms of training and prediction times. Although kernel based non-linear separators may sometimes obtain better prediction performance, compared to linear models, the speed penalty at prediction time is often substantial, since the number of support patterns often grows linearly with the size of the training set. We therefore restrict ourselves to linear models, but allow for a restricted family of explicit non-linear mappings by feature combinations.

For sentence level hedge detection in the biological domain, we employ an L_1-regularised support vector machine with hinge loss, as provided by the library implemented by Fan et al. (2008), while for weasel detection in the Wikipedia domain, we instead use the L_2-regularised maximum margin model described in more detail in section 3.1. In both cases, we approximately optimise the F_1-measure by weighting each class by the inverse of its proportion in the training data.

The reason for using L_1-regularisation in the biological domain is that the annotation is heavily biased towards a rather small number of lexical cues, making most of the potential surface features irrelevant. The Wikipedia weasel annotation, on the other hand, is much more noisy and less determined by specific lexical markers. Regularising with respect to the L_1-norm is known to give preference to sparse models and for the special case of logistic regression, Ng (2004) proved that the sample complexity grows only logarithmically in the number of irrelevant features, instead of linearly as when regularising with respect to the L_2-norm. Our preliminary experiments indicated that L_1-regularisation is superior to L_2-regularisation in the biological domain, while slightly inferior in the Wikipedia domain.

2.2 Feature Definitions

The asymmetric relationship between certain and uncertain sentences becomes evident when one tries to learn this distinction based on surface level cues. While the UNCERTAIN category is to a large extent explicitly anchored in lexical markers, the CERTAIN category is more or less defined implicitly as the complement of the UNCERTAIN category. To handle this situation, we use a bias feature to model the weight of the CERTAIN category, while explicit features are used to model the UNCERTAIN category.

The following list describes the feature templates explored for sentence level uncertainty detection. Some features are based on a linguistic analysis by the Connexor Functional Dependency (FDG) parser (Tapanainen and Järvinen, 1997).

SENLEN Preliminary experiments indicated that taking sentence length into account is beneficial. We incorporate this by using three different bias terms, according to the length (in tokens) of the sentences. This feature takes the following values: $\text{S} < 18 \leq \text{M} \leq 32 < \text{L}$.

DOCPT Document part, e.g., TITLE, ABSTRACT and BODY TEXT, allowing for different models for different document parts.

TOKEN, LEMMA Tokens in most cases equals words, but may in some special cases also be multiword units, e.g. *of course*, as defined by the FDG tokenisation. Lemmas are base forms of words, with some special features introduced for numeric tokens, e.g., year, short number, and long number.

QUANT Syntactic function of a noun phrase with a quantifier head (*at least* ***some of the isoforms*** *are conserved between mouse and humans*), or a modifying quantifier (*Recently,* ***many investigators*** *have been interested in the study on eosinophil biology*).

HEAD, DEPREL Functional dependency head of the token, and the type of dependency relation between the head and the token, respectively.

SYN Phrase-level and clause-level syntactic functions of a word.

MORPH Part-of-speech and morphological traits of a word.

Each feature template defines a set of features when applied to data. The TOKEN, LEMMA, QUANT, HEAD, DEPREL templates yield singleton sets of features for each token, while the SYN and MORPH templates extends to sets consisting of several features for each token. A sentence is represented as the union of all active token level features and the SENLEN and DOCPT, if active. In addition to the linear combination of concrete features, we allow combined features by the Cartesian product of the feature set extensions of two or more feature templates.

2.3 Feature Template Selection

Although regularised maximum margin models often cope well even in the presence of irrelevant features, it is a good idea to search the large set of potential features for an optimal subset.

In order to make this search feasible we make two simplifications. First, we do not explore the full set of individual features, but instead the set of feature templates, as defined above. Second, we perform a greedy search in which we iteratively add the feature template that gives the largest performance improvement, when added to the current optimal set of templates. The performance of a feature set for sentence level detection is measured as the mean F_1-score, with respect to the UNCERTAIN class, minus one standard deviation – the mean and standard deviation are computed by three fold cross-validation on the training set. We subtract one standard deviation from the mean in order to promote stable solutions over unstable ones.

Of course, these simplifications do not come for free. The solution of the optimisation problem might be quite unstable with respect to the optimal hyper-parameters of the learning algorithm, which in turn may depend on the feature set used. This risk could be reduced by conducting a more thorough parameter search for each candidate feature set, however, this was simply too time consuming for the present work. A further risk of using forward selection is that feature interactions are ignored. This issue is handled better with backward elimination, but that is also more time consuming.

The full set of explored feature templates is too large to be listed here; instead we list the features selected in each iteration of the search, together with their corresponding scores, in Table 1.

3 Detecting In-sentence Uncertainty

When it comes to the automatic identification of hedge cues and their linguistic scopes, Morante and Daelemans (2009) and Özgür and Radev (2009) report experiments on the BioScope corpus (Vincze et al., 2008), achieving best results (10-fold cross evaluation) on the identification of hedge cues of 71.59 F-score (using IGTree with current, preceding and subsequent word and cur-

Task	Template set	Dev F_1	Test F_1
Bio	SenLen	-	-
	∪ Lemma	88.9 (.25)	78.79
	∪ LemmaBi	90.3 (.19)	85.86
	∪ Lemma⊗Quant	90.3 (.07)	85.97
Wiki	SenLen	-	-
	∪ Token⊗DocPt	59.0 (.76)	60.12
	∪ TokenBi⊗SenLen	59.9 (.09)	58.26

Table 1: Top feature templates for sentence level hedge and weasel detection.

rent lemma as features) and 82.82 F-score (using a Support Vector Machine classifier and a complex feature set including keyword and dependency relation information), respectively. On the task of automatic scope resolution, best results are reported as 59.66 (F-score) and 61.13 (accuracy), respectively, on the full paper subset. Özgür and Radev (2009) use a rule-based method for this subtask, while Morante and Daelemans (2009) use three different classifiers as input to a CRF-based meta-learner, with a complex set of features, including hedge cue information, current and surrounding token information, distance information and location information.

3.1 Learning and Optimisation Framework

In recent years, a wide range of different approaches to general structured prediction problems, of which sequence labelling is a special case, have been suggested. Among others, Conditional Random Fields (Lafferty et al., 2001), Max-Margin Markov Networks (Taskar et al., 2003), and Structured Support Vector Machines (Tsochantaridis et al., 2005). A drawback of these approaches is that they are all quite computationally demanding. As an alternative, we propose a much more computationally lenient approach based on the regularised margin-rescaling formulation of Taskar et al. (2003), which we instead optimise by stochastic subgradient descent as suggested by Ratliff et al. (2007). In addition we only perform approximate decoding, using beam search, which allows arbitrary complex joint feature maps to be employed, without sacrificing speed.

3.1.1 Technical Details

Let $\mathcal{X}$ denote the pattern set and let $\mathcal{Y}$ denote the set of structured labels. Let $\mathcal{A}$ denote the set of atomic labels and let each label $y \in \mathcal{Y}$ consist of an indexed sequence of atomic labels $y_i \in \mathcal{A}$. Denote by $\mathcal{Y}_x \subseteq \mathcal{Y}$ the set of possible label assignments to pattern $x \in \mathcal{X}$ and by $y_x \in \mathcal{Y}_x$ its correct label. In the specific case of BIO-sequence labelling, $\mathcal{A} = \{\text{BEGIN}, \text{INSIDE}, \text{OUTSIDE}\}$ and $\mathcal{Y}_x = \mathcal{A}^{|x|}$, where $|x|$ is the length of the sequence $x \in \mathcal{X}$.

A structured classification problem amounts to learning a mapping from patterns to labels, $f : \mathcal{X} \mapsto \mathcal{Y}$, such that the expected loss $\mathrm{E}_{\mathcal{X}\times\mathcal{Y}}[\Delta(y_x, f(x))]$ is minimised. The prediction loss, $\Delta : \mathcal{Y} \times \mathcal{Y} \mapsto \Re^+$, measures the loss of predicting label $y = f(x)$ when the correct label is y_x, with $\Delta(y_x, y_x) = 0$. Here we assume the Hamming loss, $\Delta_H(y, y') = \sum_{i=1}^{|y|} \delta(y_i, y'_i)$, where $\delta(y_i, y'_i) = 1$ if $y_i \neq y'_i$ and 0 otherwise.

The idea of the margin-rescaling approach is to let the *structured margin* between the correct label y_x and a hypothesis $y \in \mathcal{Y}_x$ scale linearly with the prediction loss $\Delta(y_x, y)$ (Taskar et al., 2003). The structured margin is defined in terms of a score function $S : \mathcal{X} \times \mathcal{Y} \mapsto \Re$, in our case the linear score function $S(x, y) = \boldsymbol{w}^T\Phi(x, y)$, where $\boldsymbol{w} \in \Re^m$ is a vector of parameters and $\Phi : \mathcal{X}\times\mathcal{Y} \mapsto \Re^m$ is a joint feature function. The learning problem then amounts to finding parameters $\boldsymbol{w}$ such that $S(x, y_x) \geq S(x, y) + \Delta(y_x, y)$ for all $y \in \mathcal{Y}_x \setminus \{y_x\}$ over the training data $\mathcal{D}$. In other words, we want the score of the correct label to be higher than the score *plus the loss*, of all other labels, for each instance. In order to balance margin maximisation and margin violation, we add the L_2-regularisation term $\|\boldsymbol{w}\|^2$.

By making use of the *loss augmented* decoding function

$$f_\Delta(x, y_x) = \operatorname*{argmax}_{y \in \mathcal{Y}_x} \left[S(x, y) + \Delta(y_x, y)\right], \quad (1)$$

we get the following regularised risk functional:

$$Q_{\lambda,\mathcal{D}}(\boldsymbol{w}) = \sum_{i=1}^{|\mathcal{D}|} S_\Delta(x^{(i)}, y_{x^{(i)}}) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2, \quad (2)$$

where

$$S_\Delta(x, y_x) = \max_{y \in \mathcal{Y}_x} \left[S(x, y) + \Delta(y_x, y)\right] - S(x, y_x) \quad (3)$$

We optimise (2) by stochastic approximate subgradient descent with step size sequence $[\eta_0/\sqrt{t}]_{t=1}^{\infty}$ (Ratliff et al., 2007). The initial step size η_0 and the regularisation factor λ are data dependent hyper-parameters, which we tune by cross-validation.

This framework is highly efficient both at learning and prediction time. Training cues and scopes on the biological data, takes about a minute, while prediction times are in the order of seconds, using a Java based implementation on a standard laptop; the absolute majority of that time is spent on reading and extracting features from an inefficient internal JSON-based format.

3.1.2 Hashed Feature Functions

Joint feature functions enable encoding of dependencies between labels and relations between pattern and label. Most feature templates are defined based on input only, while some are defined with respect to output features as well. Let $\Psi(x, y_{1:i-1}, i) \in \Re^m$ denote the joint feature function corresponding to the application of all active feature templates to pattern $x \in \mathcal{X}$ and partially decoded label $y_{1:i-1} \in \mathcal{A}^{i-1}$ when decoding at position i. The feature mapping used in scoring candidate label $y_i \in \mathcal{A}$ is then computed as the Cartesian product $\Phi(x, y, i) = \Psi(x, y_{1:i-1}, i) \otimes \Lambda(y_i)$, where $\Lambda(y_i) \in \Re^m$ is a unique unitary feature vector representation of label y_i. The feature representation for a complete sequence x and its associated label y is then computed as

$$\Phi(x, y) = \sum_{i=1}^{|x|} \Phi(x, y, i)$$

When employing joint feature functions and combined features, the number of unique features may grow very large. This is a problem when the amount of internal memory is limited. Feature hashing, as described by Weinberger et al. (2009), is a simple trick to circumvent this problem. Assume that we have an original feature function $\phi : \mathcal{X} \times \mathcal{Y} \mapsto \Re^m$, where m might be arbitrarily large. Let $h : \mathbb{N}^+ \mapsto [1, n]$ be a hash function and let $h^{-1}(i) \subseteq [1, m]$ be the set of integers such that $j \in h^{-1}(i)$ iff $h(j) = i$. We now use this hash function to map the index of each feature in $\phi(x, y)$ to its corresponding index in $\Phi(x, y)$, as $\Phi_i(x, y) = \sum_{j \in h^{-1}(i)} \phi_j(x, y)$. The features in Φ are thus unions of multisets of features in ϕ. Given a hash function with good collision properties, we can expect that the subset of features mapped to any index in $\Phi(x, y)$ is small and composed of elements drawn at random from $\phi(x, y)$. Weinberger et al. (2009) contains proofs of bounds on these distributions. Furthermore, by using a k-valued hash function $h : \mathbb{N}^k \mapsto [1, n]$, the Cartesian product of k feature sets can be computed much more efficiently, compared to using a dictionary.

3.2 Position Based Feature Definitions

For in-sentence cue and scope prediction we make use of the same token level feature templates as for sentence level detection. An additional level of expressivity is added in that each token level template is associated with a token position. A template is addressed either relative to the token currently being decoded, or by the dependency arc of a token, which in turn is addressed by a relative position. The addressing can be either to a single position, or a range of positions. Feature templates may further be defined with respect to features of the input pattern, the token level labels predicted so far, or with respect to combinations of input and label features. Joint features, just as complex feature combinations, are created by forming the Cartesian product of an input feature set and a label feature set.

The feature templates are instantiated by prefixing the template name to each member of the feature set. To exemplify, the single position template TOKEN$_i$, given that the token currently being decoded at position i is *suggests*, is instantiated as the singleton set $\{\text{TOKEN}_i = \text{suggests}\}$. The range template TOKEN$_{i,i+1}$, given that the current token is *suggests* and the next token is *that*, is instantiated as the set $\{\text{TOKEN}_{i,i+1} = \text{suggests}, \text{TOKEN}_{i,i+1} = \text{that}\}$; i.e. each member of the set is prefixed by the range template name.

In addition to the token level templates used for sentence level prediction, the following templates were explored:

LABEL Label predicted so far at the addressed position(s).

HEAD.X An arbitrary feature, X, addressed by following the dependency arc(s) from the addressed position(s). For example, HEAD.LEMMA$_i$ corresponds to the lemma found by looking at the dependency head of the current token.

CUE, CUESCOPE Whether the token(s) addressed is respectively, a cue marker, or within the syntactic scope of the current cue, following the definition of scope provided by Vincze et al. (2008).

3.3 Feature Template Selection

Just as with sentence level detection, we used a greedy forward selection strategy when searching for the optimal subset of feature templates. The cue and scope detection subtasks were optimised separately.

The scoring measures used in the search for cue and scope detection features differ. In order to match the official scoring measure for cue detection, we optimise the F_1-score of labels corresponding to cue tags, i.e. we treat the BEGIN and INSIDE cue tags as an equivalence class. The official scoring measure for scope prediction, on the other hand, corresponds to the exact match of scope boundaries. Unfortunately using exact match performance turned out to be not very well suited for use in greedy forward selection. This is because before a sufficient per token accuracy has been reached, and even when it has, the exact match score may fluctuate wildly. Therefore, as a substitute, we instead guide the search by token level accuracy. This discrepancy between the search criterion and the official scoring metric is unfortunate.

Again, when taking into account position addressing, joint features and combined features, the complete set of explored templates is too large to fit in the current experiment. The selected features together with their corresponding scores are found in Table 2.

Task	Template set	Dev F_1	Test F_1
Cue	TOKEN_i	74.0 (1.5)	-
	$\cup\ \text{TOKEN}_{i-1}$	81.0 (.30)	68.78
	$\cup\ \text{MORPH}_i$	83.6 (.10)	74.06
	$\cup\ \text{LEMMA}_i \otimes \text{LEMMA}_{i+1}$	85.6 (.20)	78.41
	$\cup\ \text{SYN}_i$	86.5 (.41)	78.28
	$\cup\ \text{LEMMA}_{i-1} \otimes \text{LEMMA}_i$	86.7 (.42)	78.52
Scope	CueScope_i	66.9 (.92)	-
	$\cup\ \text{LABEL}_{i-2,i-1}$	79.5 (.67)	34.80
	$\cup\ \text{LEMMA}_i$	82.4 (1.1)	33.18
	$\cup\ \text{MORPH}_i$	83.1 (.35)	35.70
	$\cup\ \text{CUE}_{i-2,i-1}$	83.4 (.13)	40.14
	$\cup\ \text{CUE}_{i,i+1,i+2}$	83.6 (.11)	41.15
	$\cup\ \text{LEMMA}_{i-1}$	84.1 (.16)	40.04
	$\cup\ \text{MORPH}_i$	84.4 (.33)	40.04
	$\cup\ \text{TOKEN}_{i+1}$	84.5 (.09)	39.64

Table 2: Top feature templates for in-sentence detection of hedge cues and scopes.

4 Discussion

Our final F_1-score results for the corrected system are, in Task 1 for the biological domain 85.97, for the Wikipedia domain 58.25; for Task 2, our results are 39.64 for the entire task with a score of 78.52 for cue detection.

Any gold standard-based shared experiment unavoidably invites discussion on the reliability of the gold standard. It is easy to find borderline examples in the evaluation corpus, e.g. sentences that may just as well be labeled "certain" rather than "uncertain". This gives an indication of the true complexity of assessing the hidden variable of uncertainty and coercing it to a binary judgment rather than a dimensional one. It is unlikely that everyone will agree on a binary judgment every time.

To improve experimental results and the generalisability of the results for the task of detecting uncertain information on a sentence level, we would need to break reliance on the purely lexical cues. For instance, we now have identified *possible* and *putative* as markers for uncertainty, but in many instances they are not (*Finally, we wish to ensure that others can use and evaluate the GREC as simply as possible*). This would be avoidable through either a deeper analysis of the sentence to note that *possible* in this case does not modify anything of substance in the sentence, or alternatively through a multi-word term preprocessor to identify *as simply as possible* as an analysis unit.

In the Wikipedia experiment, where the objective is to identify *weasel* phrases, the judicious encoding of quantifiers such as "some of the most well-known researchers say that X" would be likely to identify the sought-for sentences when the quantified NP is in subject position. In our experiment we find that our dependency analysis did not distinguish between the various syntactic roles of quantified NPs. As a result, we marked several sentences with a quantifier as a "weasel" sentence, even where the quantified NP was in a non-subject role – leading to overly many weasel sentences. An example is given in Table 3.

If certainty can be identified separately, not as absence of overt uncertainty, identifying uncertainty can potentially be aided through the identification of explicit certainty together with negation, as found by Kilicoglu and Bergler (2008). In keeping with their results, we found negations in a sizeable proportion of the annotated training material. Currently we capture negation as a lexical cue in immediate bigrams, but with longer range negations, we will miss some clear cases: Table 3 gives two examples. To avoid these misses, we will both need to identify overt expressions of certainty and to identify and track the scope of negation – the first challenge is unexplored but would not seem to be overly complex; the second is a well-known and established challenge for NLP systems in general.

In the task of detecting in-sentence uncertainty – identification of hedge cues and their scopes – we find that an evaluation method based on exact match of a token sequence is overly unforgiving. There are many cases where the marginal tokens of a sequence are less than central or irrelevant for the understanding of the hedge cue and its scope: moving the boundary by one position over an uninteresting token may completely invalidate an otherwise arguably correct analysis. A token-by-token scoring would be a more functional evaluation criterion, or perhaps a fuzzy match, allowing for a certain amount of erroneous characters.

For our experiments, this has posed some challenges. While we model the in-sentence uncertainty detection as a sequence labelling problem in the BIO-representation (BEGIN, INSIDE, OUTSIDE), the provided corpus uses an XML-representation. Moreover, the official scoring tool requires that the predictions are well formed XML, necessitating a conversion from XML to BIO prior to training and from BIO to XML after prediction. Consistent tokenisation is important, but the syntactic analysis components used by us distorted the original tokenisation and restoring the exact same token sequence proved problematic.

Conversion from BIO to XML is straightforward for cues, while some care must be taken when annotating scopes, since erroneous scope predictions may result in malformed XML. When adding the scope annotation, we use a stack based algorithm. For each sentence, we simultaneously traverse the scope-sequence corresponding to each cue, left to right, token by token. The stack is used to ensure that scopes are either separated or nested and an additional restriction ensures that scopes may never start or end inside a cue. In case the algorithm fails to place a scope according to these restrictions, we fall back and let the scope cover the whole sentence. Several of the more frequent errors in our analyses are scoping errors, many likely to do with the fallback solution. Our analysis quite frequently fails also to assign the subject of a sentence to the scope of a hedging verb. Table 3 shows one example each of these errors – overextended scope and missing subject.

Unfortunately, the tokenisation output by our analysis components is not always consistent with the tokenisation assumed by the BioScope annotation. A post-processing step was therefore added in which each, possibly complex, token in the predicted BIO-sequence is heuristically mapped to its corresponding position in the XML structure. This post-processing is not perfect and scopes and cues at non-word token boundaries, such as parentheses, are quite often misplaced with respect to the BioScope annotation. Table 3 gives one example which is scored “erroneous” since the token “(63)” is in scope, where the “correct” solution has it outside the scope. These errors are not important to address, but are quite frequent in our results – approximately 80 errors are of this type.

To achieve more general and effective methods to detect uncertainty in an argument, we should note that uncertainty is signalled in a text through many mechanisms, and that the purely lexical and explicit signal found through the present experiments in hedge identification is effective and useful, but will not catch everything we might want to find. Lexical approaches are also domain dependent. For instance, Szarvas (2008) and Morante and Daelemans (2009) report loss in performance, when applying the same methods developed on biological data, on clinical text. Using the systems developed for scientific text elsewhere poses a migration challenge. It would be desirable both to automatically learn a hedging lexicon from a general seed set and to have features on a higher level of abstraction.

Our main result is that casting this task as a sequence labelling problem affords us the possibility to combine linguistic analyses with a highly efficient implementation of a max-margin prediction algorithm. Our framework processes the data sets in minutes for training and seconds for prediction on a standard personal computer.

5 Acknowledgements

The authors would like to thank Joakim Nivre for feedback in earlier stages of this work. This work was funded by The Swedish National Graduate School of Language Technology and by the Swedish Research Council.

References

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Richárd Farkas, Veronika Vincze, György Móra, János

Neg + certain	However, how IFN-γ and IL-4 inhibit IL-17 production is **not** yet **known**.
Neg + certain	The mechanism by which Tregs preserve peripheral tolerance is still **not** entirely **clear**.
“some”: not weasel	Tourist folks usually visit this peaceful paradise to enjoy **some leisure**$_{nonsubj}$.
“some”: weasel	**Some**$_{subj}$ suggest that the origin of music likely stems from naturally occurring sounds and rhythms.
Prediction	dRas85DV12 <xcope _.1><cue _.1>may</cue> be more potent than dEGFRλ because dRas85DV12 can activate endogenous PI3K signaling [16]</xcope>.
Gold standard	dRas85DV12 <xcope _.1><cue _.1>may</cue> be more potent than dEGFRλ</xcope> because dRas85DV12 can activate endogenous PI3K signaling [16].
Prediction	However, the precise molecular mechanisms of Stat3-mediated expression of RORγt <xcope _.1>are still <cue _.1>unclear</cue></xcope>.
Gold standard	However, <xcope _.1>the precise molecular mechanisms of Stat3-mediated expression of RORγt are still <cue _.1>unclear</cue></xcope>.
Prediction	Interestingly, Foxp3 <xcope _.1><cue _.1>may</cue> inhibit RORγt activity on its target genes, at least in par,t through direct interaction with RORγt (63)</xcope>.
Gold standard	Interestingly, Foxp3 <xcope _.1><cue _.1>may</cue> inhibit RORt activity on its target genes, at least in par,t through direct interaction with RORt</xcope> (63).

Table 3: Examples of erroneous analyses.

Csirik, and György Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of the 14th Conference on Computational Natural Language Learning (CoNLL-2010): Shared Task*, pages 1–12, Uppsala, Sweden, July. Association for Computational Linguistics.

Viola Ganter and Michael Strube. 2009. Finding hedges by chasing weasels: hedge detection using Wikipedia tags and shallow linguistic features. In *ACL-IJCNLP ’09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Morristown, NJ, USA. Association for Computational Linguistics.

Halil Kilicoglu and Sabine Bergler. 2008. Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC Bioinformatics*, 9.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th Int. Conf. on Machine Learning*. Morgan Kaufmann Publishers.

Marc Light, Xin Ying 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*, Boston, USA. ACL.

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*, Prague, Czech Republic. Association for Computational Linguistics.

Ben Medlock. 2008. Exploring hedge identification in biomedical literature. *Journal of Biomedical Informatics*, 41(4):636–654.

Roser Morante and Walter Daelemans. 2009. Learning the scope of hedge cues in biomedical texts. In *BioNLP ’09: Proceedings of Workshop on BioNLP*, Morristown, NJ, USA. ACL.

Andrew Y. Ng. 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML ’04: Proceedings of the 21st International Conference on Machine learning*, page 78, New York, NY, USA. ACM.

Arzucan Özgür and Dragomir R. Radev. 2009. Detecting speculations and their scopes in scientific text. In *Proceedings of 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore. ACL.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A comprehensive grammar of the English language*. Longman.

Nathan D. Ratliff, Andrew J. Bagnell, and Martin A. Zinkevich. 2007. (Online) subgradient methods for structured prediction. In *Eleventh International Conference on Artificial Intelligence and Statistics (AIStats)*.

György Szarvas. 2008. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proceedings of ACL-08: HLT*, Columbus, Ohio. ACL.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*.

Benjamin Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin Markov networks. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *NIPS*. MIT Press.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484.

Veronika Vincze, György Szarvas, Richárd Farkas, György Móra, and János Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(S-11).

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *ICML ’09: Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA. ACM.