UiO$_1$: Constituent-Based Discriminative Ranking for Negation Resolution

Jonathon Read **Erik Velldal** **Lilja Øvrelid** **Stephan Oepen**
University of Oslo, Department of Informatics
{jread,erikve,liljao,oe}@ifi.uio.no

Abstract

This paper describes the first of two systems submitted from the University of Oslo (UiO) to the 2012 *SEM Shared Task on resolving negation. Our submission is an adaption of the negation system of Velldal et al. (2012), which combines SVM cue classification with SVM-based ranking of syntactic constituents for scope resolution. The approach further extends our prior work in that we also identify factual negated events. While submitted for the closed track, the system was the top performer in the shared task overall.

1 Introduction

The First Joint Conference on Lexical and Computational Semantics (*SEM 2012) hosts a shared task on resolving negation (Morante and Blanco, 2012). This involves the subtasks of (i) identifying *negation cues*, (ii) identifying the in-sentence *scope* of these cues, and (iii) identifying negated (and factual) *events*. This paper describes a system submitted by the Language Technology Group at the University of Oslo (UiO). Our starting point is the negation system developed by Velldal et al. (2012) for the domain of biomedical texts, an SVM-based system for classifying cues and ranking syntactic constituents to resolve cue scopes. However, we extend and adapt this system in several important respects, such as in terms of the underlying linguistic formalisms that are used, the textual domain, handling of morphological cues and discontinuous scopes, and in that the current system also identifies negated events.

The data sets used for the shared task include the following, all based on negation-annotated Conan Doyle (CD) stories (Morante and Daelemans, 2012): a training set of 3644 sentences (hereafter referred to as CDT), a development set of 787 sentences (CDD), and a held-out evaluation set of 1089 sentences (CDE). We will refer to the combination of CDT and CDD as CDTD. An example of an annotated sentence is shown in (1) below, where the cue is marked in bold, the scope is underlined, and the event marked in italics.

(1) There was **no** *answer*.

We describe two different system configurations, both of which were submitted for the closed track (hence we can only make use of the data provided by the task organizers). The systems only differ with respect to how they were optimized. In the first configuration, (hereafter I), all components in the pipeline had their parameters tuned by 10-fold cross-validation across CDTD. The second configuration (II) is tuned against CDD using CDT for training. The rationale for this strategy is to guard against possible overfitting effects that could result from either optimization scheme, given the limited size of the data sets. For the held-out testing all models are estimated on the entire CDTD.

Unless otherwise noted, all reported scores are generated using the evaluation script provided by the organizers, which breaks down performance with respect to cues, events, scope tokens, and two variants of scope-level exact match (one requiring exact match of cues and the other only partial cue match). The latter two scores are identical for our system hence are not duplicated in this paper. Furthermore, as we did not optimize for the scope tokens measure this is only reported for the final evaluation.

Note also that the evaluation actually includes two variants of the metrics mentioned above; a set of primary measures with precision computed as $P = TP/(TP + FP)$ and a set of so-called *B measures* that instead uses $P = TP/S$, where S is the

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 310–318,
Montréal, Canada, June 7-8, 2012.

total number of predictions made by the system. The reason why S is not identical with $TP + FP$ is that partial matches are only counted as FNs (and not FPs) in order to avoid double penalties. We do not report the B measures for development testing as they were only introduced for the final evaluation and hence were not considered in our system optimization. We note though, that the relative-ranking of participating systems for the primary and B measures is identical, and that the correlation between the paired lists of scores is nearly perfect ($r = 0.997$).

The paper is structured according to the components of our system. Section 2 details the process of identifying instances of negation through the disambiguation of known cue words and affixes. Section 3 describes our hybrid approach to scope resolution, which utilizes both heuristic and data-driven methods to select syntactic constituents. Section 4 discusses our event detection component, which first applies a classifier to filter out non-factual events and then uses a learned ranking function to select events among in-scope tokens. End-to-end results are presented in Section 5.

2 Cue Detection

Cue identification is based on the light-weight classification scheme presented by Velldal et al. (2012). By treating the set of cue words as a closed class, Velldal et al. (2012) showed that one could greatly reduce the number of examples presented to the learner, and correspondingly the number of features, while at the same time improving performance. This means that the classifier only attempts to 'disambiguate' known cue words, while ignoring any words not observed as cues in the training data.

The classifier applied in the current submission is extended to also handle morphological or affixal negation cues, such as the prefix cue in ***im****patience*, the infix in *care****less****ness*, and the suffix of *colour****less***. The negation affixes observed in CDTD are; the prefixes *un*, *dis*, *ir*, *im*, and *in*; the infix *less* (we internally treat this as the suffixes *lessly* and *lessness*); and the suffix *less*. Of the total set of 1157 cues in the training and development data, 192 are affixal. There are, however, a total of 1127 tokens matching one of the affix patterns above, and while we maintain the closed class assumption also for the affixes, the classifier will need to consider their status as a cue or non-cue when attaching to any such token, as in *image*, *recklessness*, and *bless*.

2.1 Features

In the initial formulation of Velldal (2011), an SVM classifier was applied using simple n-gram features over words, both full forms and lemmas, to the left and right of the candidate cues. In addition to these token-level features, the classifier we apply here includes features specifically targeting affixal cues. The first such feature records character n-grams from both the beginning and end of the base that an affix attaches to (up to five positions). For a context like ***im****possible* we would record n-grams such as {*possi*, *poss*, ...} and {*sible*, *ible*, ...}, and combine this with information about the affix itself (*im*) and the token part-of-speech ("JJ").

For the second type of affix-specific features, we try to emulate the effect of a lexicon look-up of the remaining substring that an affix attaches to, checking its status as an independent base form and its part-of-speech. In order to take advantage of such information while staying within the confines of the closed track, we automatically generate a lexicon from the training data, counting the instances of each PoS tagged lemma in addition to n-grams of word-initial characters (again recording up to five positions). For a given match of an affix pattern, a feature will then record these counts for the substring it attaches to. The rationale for this feature is that the occurrence of a substring such as *un* in a token such as *underlying* should be less likely as a cue given that the first part of the remaining string (e.g., *derly*) would be an unlikely way to begin a word.

It is also possible for a negation cue to span multiple tokens, such as the (discontinuous) pair *neither* / *nor* or fixed expressions like *on the contrary*. There are, however, only 16 instances of such multiword cues (MWCs) in the entire CDTD. Rather than letting the classifier be sensitive to these corner cases, we cover such MWC patterns using a small set of simple post-processing heuristics. A small stop-list is used for filtering out the relevant words from the examples presented to the classifier (*on*, *the*, etc.). Note that, in terms of training the final classifiers, CDTD provides us with a total of 1162 positive and

Data set	Model	Prec	Rec	F_1
CDTD	Baseline	92.25	88.50	90.34
	$\text{Classifier}_{\text{I}}$	94.99	95.07	95.03
CDD	Baseline	90.68	84.39	87.42
	$\text{Classifier}_{\text{II}}$	93.75	95.38	94.56
CDE	Baseline	87.10	92.05	89.51
	$\text{Classifier}_{\text{I}}$	91.42	92.80	92.10
	$\text{Classifier}_{\text{II}}$	89.17	93.56	91.31

Table 1: Detecting negation cues using the two classifiers and the majority-usage baseline.

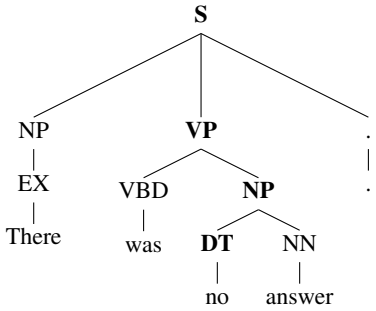

Figure 1: Example parse tree provided in the data, highlighting our candidate scope constituents.

1100 negative training examples, given our closed-class treatment of cues.

Before we turn to the results, note that the difference between the two submitted versions of the classifier (I and II) only concerns the orders of the n-grams used for the token-level features.[1]

2.2 Results

Table 1 presents the results for our cue classifier. As an informed baseline, we also tried classifying each word based on its most frequent use as a cue or non-cue in the training data. (Affixal cue occurrences are counted by looking at both the affix-pattern and the base it attaches to, basically treating the entire token as a cue. Tokens that end up being classified as cues are then matched against the affix patterns observed during training in order to correctly delimit the annotation of the cue.) This simple majority-usage approach actually provides a fairly strong baseline, yielding an F_1 of 90.34 on CDTD. Compare this to the F_1 of 95.03 obtained by the classifier on the same data set. However, when applying the models to the held-out set, with models estimated over the entire CDTD, the classifier suffers a slight drop in performance, leaving the baseline even more competitive: While our best performing final cue classifier (I) achieves F_1=92.10, the baseline achieves F_1=89.51, and even outperforms four of the ten cue detection systems submitted for the shared task (three of the 12 shared task submissions use the same classifier).

[1]Classifier I records the lemma and full form of the target token, and lemmas two positions left/right. Classifier II records the lemma, form, and PoS of the target, full forms three positions to the left and one to the right, PoS one position right/left, and lemmas three positions to the right. The affixal-specific features are the same for both configurations as described above.

Inspecting the predictions of the classifier on CDD, which comprises a total of 173 gold annotated cues, we find that Classifier I mislabels 11 false positives (FPs) and seven false negatives (FNs). Of the FPs, we find five so-called *false negation cues* (Morante et al., 2011), including three instances of the fixed expression *none the less*. The others are affixal cues, of which two are clearly wrong (***un****derworked*, ***un****iversal*) while others might arguably be due to annotation errors (***in****superable*, ***un****happily*, *end****less***, *list****less****ly*). Among the FNs, two are due to MWCs not covered by our heuristics (e.g., *no more*), with the remainder concerning affixes.

3 Constituent-Based Scope Resolution

During the development of our scope resolution system we have pursued both a rule-based and data-driven approach. Both are rooted in the assumption that the scope of negations corresponds to a syntactically meaningful unit. Our starting point here will be the syntactic analyses provided by the task organizers (see Figure 1), generated using the reranking parser of Charniak and Johnson (2005). However, as alignment between scope annotations and syntactic units is not straightforward for all cases, we apply several exception rules that 'slacken' the requirements for alignment, as discussed in Section 3.1. In Sections 3.2 and 3.3 we detail our rule-based and data-driven approaches, respectively. Note that the predictions of the rule-based component will be incorporated as features in the learned model, similarly to the set-up described by Read et al. (2011). Section 3.4 details the post-processing we apply to handle cases of discontinuous scope, be-

fore Section 3.5 finally presents development results together with a brief error analysis.

3.1 Constituent Alignment and Slackening

In order to test our initial assumption that syntactic units correspond to scope annotations, we quantify the alignment of scopes with constituents in CDT, excluding 97 negations that do not have a scope. We find that the initial alignment is rather low at 52.42%. We therefore formulate a set of *slackening* heuristics, designed to improve on this alignment by removing certain constituents at the beginning or end of a scope. First of all, removing constituent-initial and -final punctuation improves alignment to 72.83%. We then apply the following slackening rules, with examples indicating the resulting scope following slackening (not showing events):

- Remove coordination (CC) and following conjuncts if the coordination is a rightwards sibling of an ancestor of the cue and it is not directly dominated by an NP.

  (2) Since we have been so **un***fortunate* as to miss him and have no notion [...]

- Remove S* to the right of cue, if delimited by punctuation.

  (3) "There is **no** other *claimant*, I presume ?"

- Remove constituent-initial SBAR.

  (4) If it concerned no one but myself I would **not** try to keep it from you."

- Remove punctuation-delimited NPs.

  (5) "But I *ca***n't** forget them, Miss Stapleton," said I.

- Remove constituent-initial RB, CC, UH, ADVP or INTJ.

  (6) And yet it was **not** quite the *last*.

The slackening rules are based on a few observations. First, scope rarely crosses coordination boundaries (with the exception of nominal coordination). Second, scope usually does not cross clause boundaries (indicated by S/SBAR). Furthermore, titles and other nominals of address are not included in the scope. Finally, sentence and discourse adverbials are often excluded from the scope. Since these express semantic distinctions, we approximate this notion syntactically using parts-of-speech and constituent category labels expressing adverbials (RB), coordinations (CC), various types of interjections (UH, INTJ) and adverbial phrases (ADVP). We may note here that syntactic categories are not always sufficient to express semantic distinctions. Prepositional phrases, for instance, are often used to express the same type of discourse adverbials, but can also express a range of other distinctions (e.g., temporal or locative adverbials), which *are* included in the scope. So a slackening rule removing initial PPs was tried but not found to improve overall alignment.

```
RB//VP/SBAR if SBAR\WH*
RB//VP/S
RB//S
DT/NP if NP/PP
DT//SBAR if SBAR\WHADVP
DT//S
JJ//ADJPVP/S if S\VP\VB*[@lemma="be"]
JJ/NP/NP if NP\PP
JJ//NP
UH
IN/PP
NN/NP//S/SBAR if SBAR\WHNP
NN/NP//S
CC/SINV
```

Figure 2: Scope resolution heuristics.

After applying the above slackening rules the alignment rate for CDT improves to 86.13%. This also represents an upper-bound on our performance, as we will not be able to correctly predict a scope that does not align with a (slackened) constituent.

3.2 Heuristics Operating over Constituents

The alignment of constituents and scopes reveal consistent patterns and we therefore formulate a set of heuristic rules over constituents. These are based on frequencies of paths from the cue to the scope-aligned constituent for the annotations in CDT, as well as the annotation guidelines (Morante et al., 2011). The rules are formulated as paths over constituent trees and are presented in Figure 2. The path syntax is based on LPath (Lai and Bird, 2010). The rules are listed in order of execution, showing how more specific rules are consulted before more general ones. We furthermore allow for some additional functionality in the interpretation of rules by enabling simple constraints that are applied to the candidate constituent. For example, the rule `RB//VP/SBAR if SBAR\WH*` will be activated when the cue is an adverb having some ancestor VP which has a parent SBAR, where the SBAR must contain a WH-phrase among its children.

In cases where no rule is activated we use a *default scope* prediction, which expands the scope to both the left and the right of the cue until either the sentence boundary or a punctuation mark is reached. The rules are evaluated individually in Section 3.5 below and the rule predictions are furthermore employed as features for the ranker described below.

3.3 Constituent Ranking

Our data-driven approach to scope resolution involves learning a ranking function over candidate syntactic constituents. The approach has similarities to discriminative parse selection, except that we here rank subtrees rather than full parses.

When defining the training data, we begin by selecting negations for which the parse tree contains a constituent that (after slackening) aligns with the gold scope. We then select an initial candidate by selecting the smallest constituent that spans all the words in the cue, and then generate subsequent candidates by traversing the path to the root of the tree (see Figure 1). This results in a mean ambiguity of 4.9 candidate constituents per negation (in CDTD). Candidates whose projection corresponds to the gold scope are labeled as correct; all others are labeled as incorrect. Experimenting with a variety of feature types (listed in Table 2), we use the implementation of ordinal ranking in the SVM^{light} toolkit (Joachims, 2002) to learn a linear scoring function for preferring correct candidate scopes.

The most informative feature type is the *LPath from cue*, which in addition to recording the full path from the cue to the candidate constituent (e.g., the path to the correct candidate in Figure 1 is `no/DT/NP/VP/S`), also includes delexicalized (`./DT/NP/VP/S`), generalized (`no/DT//S`), and generalized delexicalized versions (`./DT//S`).

Note that the *rule prediction* feature facilitates a hybrid approach by recording whether the candidate matches the boundaries of the scope predicted by the rules of Section 3.2, as well as the degree of overlap.

3.4 Handling Discontinuous Scope

10.3% of the scopes in the training data are what (Morante et al., 2011) refer to as *discontinuous*. This means that the scope contains two or more parts which are bridged by tokens other than the cue.

Feature types	**I**	**II**
LPath from cue	•	•
LPath from cue bigrams and trigrams	•	•
LPath from cue to left/right boundary	•	
LPath to left/right boundary		•
LPath to root	•	
Punctuation to left/right	•	•
Rule prediction		•
Sibling bigrams		•
Size in tokens, relative to sentence (%)	•	•
Surface bigrams	•	•
Tree distance from cue	•	•

Table 2: Features used to describe candidate constituents for scope resolution, with indications of presence in our two system configurations.

(7) I therefore spent the day at my club and did **not** return to Baker Street until evening.

(8) There was certainly **no** physical injury of any kind.

The sentence in (7) exemplifies a common cause of scopal discontinuity in the data, namely ellipsis (Morante et al., 2011). Almost all of these are cases of coordination, as in example (7) where the cue is found in the final conjunct (*did not return [. . .]*) and the scope excludes the preceding conjunct(s) (*therefore spent the day at my club*). There are also some cases of adverbs that are excluded from the scope, causing discontinuity, as in (8), where the adverb *certainly* is excluded from the scope.

In order to deal with discontinuous scopes we formulate two simple post-processing heuristics, which are applied after rules/ranking: (1) If the cue is in a conjoined phrase, remove the previous conjuncts from the scope, and (2) remove sentential adverbs from the scope (where a list of sentential adverbs was compiled from the training data).

3.5 Results

Our development procedure evaluated all permutations of feature combinations, searching for optimal parameters using gold-standard cues. Table 2 indicates which features are included in our two ranker configurations, i.e., tuning by 10-fold cross-validation on CDTD (I) vs. a train/test-split for CDT/CDD(II).

Table 3 lists the results of our scope resolution approaches applied to gold cues. As a baseline, all

Data set	Model	Prec	Rec	F_1
CDTD	Baseline	98.31	33.18	49.61
	Rules	100.00	71.37	83.29
	Ranker_{I}	100.00	73.55	84.76
CDD	Baseline	100.00	36.31	53.28
	Rules	100.00	69.64	82.10
	$\text{Ranker}_{\text{II}}$	100.00	70.24	82.52
CDE	Baseline	96.47	32.93	49.10
	Rules	98.73	62.65	76.66
	Ranker_{I}	98.77	64.26	77.86
	$\text{Ranker}_{\text{II}}$	98.75	63.45	77.26

Table 3: Scope resolution for gold cues using the two versions of the ranker, also listing the performance of the rule-based approach in isolation.

cases are assigned the default scope prediction of the rule-based approach. On CDTD this results in an F_1 of 49.61 (P=98.31, R=33.18); compare to the ranker in Configuration I on the same data set (F_1=84.76, P=100.00, R=73.55). We note that our different optimization procedures do not appear to have made much difference to the learned ranking functions as both perform similarly on the held-out data, though suffering a slight drop in performance compared to the development results. We also evaluate the rules and observe that this approach achieves similar held-out results. This is particularly note-worthy given that there are only fourteen rules plus the default scope baseline. Note that, as the rankers performed better than the rules in isolation on both CDTD and CDD during development, our final system submissions are based on rankers I and II from Table 3.

We performed a manual error analysis of our scope resolution system ($\text{Ranker}_{\text{II}}$) on the basis of CDD (using gold cues). First, we may note that parse errors are a common sources of scope resolution errors. It is well-known that coordination presents a difficult construction for syntactic parsers, and we often find incorrectly parsed coordinate structures among the system errors. Since coordination is used both in the slackening rules and the analysis of discontinuous scopes, these errors have clear effects on system performance. We may further note that discourse-level adverbials, such as *in the second place* in example (9) below, are often included in the scope assigned by our system, which they should not be according to the gold annotation.

(9) But, in the second place, why did you **not** come at once?

There are also quite a few errors related to the scope of affixal cues, which the ranker often erroneously assigns a scope that is larger than simply the base which the affix attaches to.

4 Event Detection

Our event detection component implements two stages: First we apply a factuality classifier, and then we identify negated events[2] for those contexts that have been labeled as factual. We detail the two stages in order below.

4.1 Factuality Detection

The annotation guidelines of Morante et al. (2011) specify that events should only be annotated for negations that have a scope and that occur in factual statements. This means that we can view the *SEM data sets to implicitly annotate factuality and non-factuality, and take advantage of this to train an SVM factuality classifier. We take positive examples to correspond to negations annotated with both a scope and an event, while negative examples correspond to scope negations with no event. For CDTD, this strategy gives 738 positive and 317 negative examples, spread over a total of 930 sentences. Note that we do not have any explicit annotation of cue words for these examples. All we have are instances of negation that we know to be within a factual or non-factual context, but the indication of factuality may typically be well outside the annotated negation scope. For our experiments here, we therefore use the negation cue itself as a place-holder for the abstract notion of context that we are really classifying. Given the limited amount of data, we only optimize our factuality classifier by 10-fold cross-validation on CDTD (i.e., the same configuration is used for submissions I and II).

The feature types we use are all variations over bag-of-words (BoW) features. We include left- and right-oriented BoW features centered on the negation cue, recording forms, lemmas, and PoS, and using both unigrams and bigrams. The features are ex-

[2]Note that the annotation guidelines use the term *event* rather broadly as referring to a process, action, state, or property (Morante et al., 2011).

Data set	Model	Prec	Rec	F_1	Acc
CDTD	Baseline	69.95	100.00	82.32	69.95
	Classifier	84.51	96.07	89.92	83.98
CDE	Baseline	69.48	100.00	81.99	69.48
	Classifier	77.73	95.91	85.86	78.31

Table 4: Results for factuality detection (using gold negation cues and scopes). Due to the limited training data for factuality, the classifier is only optimized by 10-fold cross-validation on CDTD.

tracted from the sentence as a whole, as well as from a local window of six tokens to each side of the cue.

Table 4 provides results for factuality classification using gold-standard cues and scopes.[3] We also include results for a baseline approach that simply considers all cases to be factual, i.e., the majority class. In this case precision is identical to accuracy and recall is 100%. For precision and accuracy we see that the classifier improves substantially over the baseline on both data sets, although there is a bit of a drop in performance when going from the 10-fold to held-out results. There also seem to be some signs of overfitting, given that roughly 70% of the training examples end up as support vectors.

4.2 Ranking Events

Having filtered out non-factual contexts, events are identified by applying a similar approach to that of the scope-resolving ranker described in Section 3.3. In this case, however, we rank tokens as candidates for events. For simplicity in this first round of development we make the assumption that all events are single words. Thus, the system will be unable to correctly predict the event in the 6.94% of instances in CDTD that are multi-word.

We select candidate words from all those marked as being in the scope (including substrings of tokens with affixal cues). This gives a mean ambiguity of 7.8 candidate events per negation (in CDTD). Then, discarding multi-word training examples, we use SVM^{light} to learn a ranking function for identifying events among the candidates.

Table 5 shows the features employed, with indications as to their presence in our two configurations (after an exhaustive search of feature combinations). The most important feature was *LPath to scope constituent*. For example, in Figure 1 the scope constituent is the `S` root of the tree; the path that describes the correct candidate is `answer/NN/NP/VP/S`. As discussed in Section 3.3, we also record generalized, delexicalized and generalized delexicalized paths.

[3] As this is not singled out as a separate subtask in the shared task itself, these are the only scores in the paper not computed using the script provided by the organizers.

Feature type	I	II
Contains affixal cue	•	
Following lemma		•
Lemma	•	•
LPath to scope constituent	•	•
LPath to scope constituent bigrams	•	•
Part-of-speech	•	•
Position in scope	•	•
Preceding lemma	•	•
Preceding part-of-speech	•	•
Token distance from cue	•	•

Table 5: Features used to describe candidates for event detection, with indications of presence in our two system configurations.

Data set	Model	Prec	Rec	F_1
CDTD	Ranker_{I}	91.49	90.83	91.16
CDD	$\text{Ranker}_{\text{II}}$	92.11	91.30	91.70
CDE	Ranker_{I}	83.73	83.73	83.73
	$\text{Ranker}_{\text{II}}$	84.94	84.95	84.94

Table 6: Event detection for gold scopes and gold factuality information.

Table 6 lists the results of the event ranker applied to gold-standard cues, scopes, and factuality. For a comparative baseline, we implemented a keyword-based approach that simply searches in-scope words for instances of events previously observed in the training set, sorted according to descending frequency. This baseline achieves F_1=29.44 on CDD. For comparison, the ranker (II) achieves F_1=91.70 on the same data set, as seen in Table 6. We also see that Configuration II appears to generalize best, with over 1.2 points improvement over the F_1 of I.

An analysis of the event predictions for CDD indicates that the most frequent errors (41.2%) are instances where the ranker correctly predicts part of the event but our single word assumption is invalid. Another apparent error is that the system fails to

	Submission I			Submission II		
	Prec	**Rec**	**F_1**	**Prec**	**Rec**	**F_1**
Cues	91.42	92.80	92.10	89.17	93.56	91.31
Scopes	87.43	61.45	72.17	83.89	60.64	70.39
Scope Tokens	81.99	88.81	85.26	75.87	90.08	82.37
Events	60.50	72.89	66.12	60.58	75.00	67.02
Full negation	83.45	43.94	57.57	79.87	45.08	57.63
Cues B	89.09	92.80	90.91	86.97	93.56	90.14
Scopes B	59.30	61.45	60.36	56.55	60.64	58.52
Events B	57.62	72.89	64.36	58.60	75.00	65.79
Full negation B	42.18	43.94	43.04	41.90	45.08	43.43

Table 7: End-to-end results on the held-out data.

predict a main verb for the event, and instead predicts nouns (17.7% of all errors), modals (17.7%) or prepositions (11.8%).

5 Held-Out Evaluation

Table 7 presents our final results for both system configurations on the held-out evaluation data (also including the B measures, as discussed in the introduction). Comparing submission I and II, we find that the latter has slightly better scores end-to-end. However, as seen throughout the paper, the picture is less clear-cut when considering the isolated performance of each component. When ranked according to the *Full Negation* measures, our submissions were placed first and second (out of seven submissions in the closed track, and twelve submissions total). It is difficult to compare system performance on subtasks, however, as each component will be affected by the performance of the previous.

6 Conclusions

This paper has presented two closed-track submissions for the *SEM 2012 shared task on negation resolution. The systems were ranked first and second overall in the shared task end-to-end evaluation, and the submissions only differ with respect to the data sets used for parameter tuning. There are four components in the pipeline: (i) An SVM classifier for identifying negation cue words and affixes, (ii) an SVM-based ranker that combines empirical evidence and manually-crafted rules to resolve the in-sentence scope of negation, (iii) a classifier for determining whether a negation is in a factual or non-factual context, and (iv) a ranker that determines (factual) negated events among in-scope tokens.

For future work we would like to try training separate classifiers for affixal and token-level cues, given that largely separate sets of features are effective for the two cases. The system might also benefit from sources of information that would place it in the open track. These include drawing information from other parsers and formalisms, generating cue features from an external lexicon, and using additional training data for factuality detection, e.g., FactBank (Saurí and Pustejovsky, 2009).

From observations on CDTD we note that approximately 14% of scopes will be unresolvable as they are not aligned with constituents (see Section 3.1). This can perhaps be tackled by ranking tokens as candidates for left and right scope boundaries (similar to the event ranker in the current work). This would improve the upper-bound to 100% at the expense of greatly increasing the number of candidates. However, the strong discriminative power of our current approach can still be incorporated using constituent-based features.

Acknowledgments

We thank Roser Morante and Eduardo Blanco for their work in organizing this shared task and commitment to producing quality data. We also thank the anonymous reviewers for their feedback. Large-scale experimentation was carried out with the TITAN HPC facilities at the University of Oslo.

References

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the Forty-Third Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, MI.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM International Conference on Knowledge Discovery and Data Mining*, Alberta.

Catherine Lai and Steven Bird. 2010. Querying linguistic trees. *Journal of Logic, Language and Information*, 19:53–73.

Roser Morante and Eduardo Blanco. 2012. *SEM 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, Montreal.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation in Conan Doyle stories. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, Istanbul.

Roser Morante, Sarah Schrauwen, and Walter Daelemans. 2011. Annotation of negation cues and their scope: Guidelines v1.0. Technical report, University of Antwerp. CLIPS: Computational Linguistics & Psycholinguistics technical report series.

Jonathon Read, Erik Velldal, Stephan Oepen, and Lilja Øvrelid. 2011. Resolving speculation and negation scope in biomedical articles using a syntactic constituent ranker. In *Proceedings of the Fourth International Symposium on Languages in Biology and Medicine*, Singapore.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227–268.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers and the role of syntax. *Computational Linguistics*, 38(2).

Erik Velldal. 2011. Predicting speculation: A simple disambiguation approach to hedge detection in biomedical literature. *Journal of Biomedical Semantics*, 2(5).