An open-source tool for negation detection: a maximum-margin approach

Martine Enger **Erik Velldal** **Lilja Øvrelid**

University of Oslo, Department of Informatics
{marenger,erikve,liljao}@ifi.uio.no

Abstract

This paper presents an open-source toolkit for negation detection. It identifies negation cues and their corresponding scope in either raw or parsed text using maximum-margin classification. The system design draws on best practice from the existing literature on negation detection, aiming for a simple and portable system that still achieves competitive performance. Pre-trained models and experimental results are provided for English.

1 Introduction

The task of negation detection has recently seen quite a bit of interest in the NLP community, in part spurred by the availability of annotated data and evaluation software introduced by the shared tasks at CoNLL 2010 (Farkas et al., 2010) and *SEM 2012 (Morante and Blanco, 2012). While many research-based systems have been developed, with the aim of exploring features and algorithms to advance the state-of-the-art in terms of performance (Morante and Daelemans, 2009; Read et al., 2012; Lapponi et al., 2012; Packard et al., 2014; Fancellu et al., 2016), many of them are difficult to employ in practice, due to layered architectures and many dependencies, and furthermore, most are simply not made publicly available in the first place.

In this paper, we present an open-source portable toolkit for automatic negation detection, with experimental results reported for English. The system is implemented in Python on top of PyStruct (Müller and Behnke, 2014), a library for structured prediction based on a maximum-margin approach. The system implements two stages of negation analysis, namely cue detection, which detects words that signal negation, such as *no*, *not* and *un*fortunate, and scope resolution, which identifies the span of the sentence that is affected by the negation. Our negation toolkit builds on existing libraries that are actively maintained and easy to install, and the source[1] is made freely available (GPL). While we make pre-trained classifiers available (for English), users will also be able to train their own models.

The system design is based on best practices from previous work, in particular systems from the 2012 *SEM shared task. In particular, we adopt the practice of solving scope resolution as a sequence labeling task (Morante and Daelemans, 2009; Lapponi et al., 2012; White, 2012) based on syntactic features (Read et al., 2012; Lapponi et al., 2012; Packard et al., 2014). In contrast to many of the previous systems that have used constituency-based representations (Read et al., 2012; Packard et al., 2014), we base our syntactic features on dependency representations, similar to the approach of Lapponi et al. (2012). For cue detection, on the other hand, simply using surface-oriented lexical features have been shown to be sufficient, and we here largely build on the specific approach described by Read et al. (2012; Velldal et al. (2012), using a binary SVM classifier.

The main goal of this work is to arrive at a lean and light-weight system with minimal use of extra heuristics beyond machine learned models. While achieving the highest performance was not our main goal, the results are competitive with previously reported SoA results in the literature. Moreover, the system can be employed with both raw and parsed input data.

2 Experimental set-up

The Conan Doyle corpus The data set we use for training and testing is the Conan Doyle cor-

[1] https://github.com/marenger/negtool

Proceedings of the Workshop Computational Semantics Beyond Events and Roles (SemBEaR), pages 64–69,
Valencia, Spain, April 4, 2017.

pus (Morante and Daelemans, 2012) as used in the 2012 *SEM shared task (Morante and Blanco, 2012), based on a CoNLL-style format. While the shared task also included detection of events and focus, we only focus on cues and scopes in this work. We use the same splits for training, development testing and held-out evaluation as supplied for the shared task. Examples (1)-(2) below show two examples taken from the corpus, where negation cues are in bold and their scopes are underlined. In (1), the cue is the adverb *not*, whereas (2) provides an example of the affixal cue *un*.

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

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

The Conan Doyle corpus provides phrase structure trees produced by the Charniak and Johnson (2005) parser, and we have used the Stanford Parser (Manning et al., 2014) to convert these to Stanford basic dependency representations (de Marneffe et al., 2014) prior to training.

Evaluation We use the evaluation script of the 2012 *SEM shared task (Morante and Blanco, 2012) for measuring precision, recall and F-score. For scopes, it provides two different measures; *token-level* and *scope-level*. For the token-level measure the evaluation is defined similarly as for cues, simply checking whether each token in the scope sequence is correctly labeled. For scopes on the other hand, a true positive requires both the entire scope sequence and cue to be correct.

Note that for the held-out results, our system is trained on both the development and training data combined.

System comparison In addition to providing baseline results for both cues and scopes, we also include the results for the UiO_2 system of Lapponi et al. (2012) from the *SEM shared task. Achieving the best results for both cue and scope resolution in the open track, it has guided much of the design of the current system. The cue classification component of UiO_2 was the same as for UiO_1 (run 1) (Read et al., 2012) – the system that was ranked first in the shared task overall (though not for cue detection in isolation).

Maximum-margin learning for cues and scopes While cue detection is here approached as a token-wise classification problem and scope resolution as sequence classification, they are both modeled using a maximum-margin approach. Cue detection is solved using a binary Support Vector Machine (SVM) classifier (Vapnik, 1995). As is fairly common, scope resolution is solved as a sequence labeling task, applying a discriminative linear-chain Conditional Random Fields (CRF) model.However, in a conventional CRF, the parameters are learned through maximum likelihood estimation. In PyStruct on the other hand, the parameters are estimated through maximum-margin learning based on SVMs, resulting in what may be called a maximum-margin CRF.

System requirements The input given to the system can either be raw running text or parsed data in the CoNLL-X format (Buchholz and Marsi, 2006). If the user inputs raw text, we need to tokenize, tag and parse the text before we can classify the sentences. Because our training data uses PTB PoS-tags and Stanford dependencies (following conversion), we need a pipeline providing the same standard, and hence use the CoreNLP tool (Manning et al., 2014). Beyond Python 2.7 or newer, the negation tool has the following dependencies: scikit-learn, PyStruct, NumPy, and NetworkX (in addition to CoreNLP unless pre-parsed input is provided).

3 Cue identification

The task of cue detection is to identify potential cue words and determine whether they function as negation cues in the given context. Cue detection is here solved using a binary SVM classifier and follows the filtering approach described by Velldal (2011) and Read et al. (2012) which means that not all words in the input text are presented to the classifier. Instead we extract a lexicon of known single-word cues from the training data and only attempt to disambiguate these (any other word will always be labeled as a non-cue). Additionally, a separate lexicon of affixal cues is also automatically extracted, consisting of affixes seen in the training data, viz. the prefixes {*dis, im, in, ir, un*}, the infix *less*, and the suffix *less*. The cue classifier is presented with any words that match either of these at the respective positions, e.g. words that have a prefix that matches any of the prefixes, e.g. *im*patient and *im*age.

In theory, this way of restricting the problem to a closed class of candidates will put a cap on the upper bound of recall. In practice, Velldal (2011) found that it could still outperform full

	Development			Held-out		
	P	**R**	**F1**	**P**	**R**	**F1**
Baseline	90.68	84.39	87.42	87.10	92.05	89.51
UiO_2	93.75	95.38	94.56	89.17	93.56	91.31
System	91.67	95.38	93.49	90.15	93.56	91.82

Table 1: Cue classification

word-by-word classification where all words are considered. It simplifies the problem in that much fewer instances need to be considered, thereby also greatly reducing the feature space, and also gives much more balanced classes.

Multi-word cues, like '*by no means*' or '*neither...nor*', are handled by a few simple post-processing rules, simply checking whether a given cue word forms part of a multi-word cue in the given context. Using a small stop-list, some forms like *by* and *means* are excluded from the list of candidate cue words considered by the classifier.

As a baseline we use a majority class classifier, labeling each word by its most frequent label in the training data. Table 1 shows that this simple baseline is already quite strong: With an F1 of 89.51 on the held-out data it outperforms 4 of the 12 systems submitted for the *SEM shared task.

The feature configuration of the cue classifier is based on a grid search towards the development set mostly based on features previously described by Read et al. (2012; Velldal et al. (2012), tuning the SVM C parameter separately for each configuration. The final model uses the following features for each token to be classified: The word form, PoS and lemma of the token, as well as lemmas $\pm$ 1 position. For candidates of affixal cues we additionally extract the affix itself and character n-grams up to n=5 of the base-form that the affix attaches to (extracted from both the beginning and the end of the form). In terms of PyStruct configuration we use its BinaryClf model with the NSlackSSVM estimator, with the C regularization parameter set to 0.2.

The results are shown in Table 1. We see that there is a slight drop in F1-score when moving from the development set to the held-out set (from 93.49 to 91.82). Compared to UiO_2, we see that while the recall of the two systems are identical, the precision of our system is almost 1 percentage point higher. Overall, our cue classifier would have ranked third in the *SEM 2012 shared task.

4 Scope resolution

Our approach to scope resolution largely follows that of the UiO_2 system of Lapponi et al. (2012) from the *SEM shared task, both in terms of the choice of machine learning algorithm, the internal data representation and the set of features used to represent the negation scopes. Like them, we model scope resolution as a sequence labeling task, making use of both lexical and syntactic information regarding the context of a negation cue. Just as for the cue classifier, we performed extensive tuning towards the development set for the maximum-margin CRF scope model – experimenting with different features, sequence labels, and hyper-parameters – finally arriving at the following configuration:

Surface features: The word form, lemma ($\pm$ 1 position), and PoS ($\pm$ 1 position)

Cue features: Cue type, left/right cue distance, and cue PoS.

Dependency features: Directed dependency distance, dependency graph path.

Note that the directed dependency path is the shortest path from the head of the cue to the current token. Internally, we employ the following label set to represent scopes: I, O, B, and C (Inside, Outside, Beginning, and Cue). Note that the only post-processing performed after the CRF is ensuring that the cue is always out-of-scope and that the base of an affixal cue is always in-scope. In terms of PyStruct configuration we use its chain CRF model, with the FrankWolfeSSVM estimator, with the C regularization parameter set to 0.1.

The results are presented in Table 2. As for the cue results in Table 1 we here too report the performance of the UiO_2 system as a point of reference. In addition, we also include results for a baseline corresponding to labeling the entire sentence as in-scope if it contains a (gold) negation cue. While this section focuses on scope prediction performance in isolation using gold cues, Section 5 discusses the end-to-end results with scope resolution for predicted cues.

We see that the baseline scores are much lower on the evaluation set than the development set, with the scope-level F-score decreasing from 32.03% on development to 19.24% on the evaluation set. However, our maximum-margin scope classifier appears to be robust to this gap, and its

	Development						Held-out					
	Scope-level			Token-level			Scope-level			Token-level		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline (gold cues)	86.84	19.64	32.03	45.00	97.55	61.59	66.67	11.24	19.24	38.54	98.01	55.32
UiO_2 (gold cues)	100.00	66.67	80.00	90.64	81.36	85.75	-	-	-	-	-	-
System (gold cues)	100.00	63.10	77.38	90.80	82.05	86.20	98.75	63.45	77.26	91.47	81.39	86.14
UiO_2	-	-	-	-	-	-	85.71	62.65	72.39	86.03	81.55	83.73
System	88.14	61.90	72.73	85.24	80.56	82.83	85.00	61.45	71.33	85.49	80.28	82.80

Table 2: Scope resolution, for both gold and predicted cues.

performance remains largely unchanged across the two test sets, with only a 0.12 percentage points decrease in F-score for the scope-level and 0.06 on the token-level.

Turning to the development results of the scope CRF model of UiO_2 (on gold cues), we find that the scores are slightly higher than ours with respect to the scope-level, but slightly lower for the token-level. For the held-out evaluation data on the other hand, UiO_2 scope results for gold cues were not reported, like for most of the other *SEM competition systems, unfortunately. However, the system description of the UiO_1 system (Read et al., 2012) – implementing a hybrid approach combining manually defined rules and SVM-based ranking of constituent (sub-)trees – reports scope-level scores for gold cues on both the development and evaluation data. The same holds for the system of Packard et al. (2014), which combines the UiO_1 system with an additional layer of manually defined rules over Minimal Recursion Semantics structures created by an HPSG parser. For both of these systems we can observe a larger drop in F1 when moving from the development data to the evaluation data, with the UiO_1/MRS-combination dropping from 82.5 to 78.7, and with the UiO_1 system[2] on its own dropping from 82.52 to 77.26 (compared to the drop from 77.38 to 77.26 in the case of our system). Regardless of the causes for these differences, it at least appears that our purely CRF-based system, with the tuning of the C parameter for the underlying maximum-margin model, does not suffer any overfitting effects. At the same time, we see that the combined system of Packard et al. (2014) achieves the highest absolute scores, and we return to this point when discussing end-to-end results below. Finally, note that Fancellu et al. (2016) report scope results on the *SEM evaluation data (gold cues only) for a suite of different classifiers based on a bi-directional LSTM, with the best configuration obtaining a scope-level F-score of 77.77. In sum, we observe two things; (i) our scope classifier achieves competetive performance, and (ii) despite the large differences in terms of types of approaches and architectures for the various scope systems considered here, there are not large differences in terms of performance.

[2]We here report results for 'run II' of UiO_1 as submitted for the *SEM 2012 shared task, since this version of the system was optimized towards the development set just like in our set-up, while 'run I' was optimized by cross-validation on the training and development data combined.

4.1 Error analysis

We performed an error analysis of our scope resolution predictions over the development data using gold cues. The analysis shows that our system struggles with discontinuous scopes, as in (3):

(3) It was **not**, I must confess, a very alluring prospect.

This is not surprising, seeing that several of the top performing systems implements dedicated post-processing modules for dealing with discontinuous scopes. The error analysis also reveals other types of recurring scope errors, including sentences that contains multiple negation cues with overlapping scopes (gold-standard). Moreover, we also observed that many of the sentences that are counted as false negatives with respect to the strict exact match scope-level measure often just have a single token that is incorrectly labeled, meaning that the overall scope is very close to being correct. This is reflected in the fact that the token-level F-score is roughly 10 percentage points higher than the scope-level F-score.

5 End-to-end results

As expected, the scope scores drop when moving from gold to predicted cues, mostly in terms of precision, which for the scope-level on the development set was 100% with gold cues but 88.14%

with predicted cues. Errors from the cue classifier propagates to the scope classifier which will attempt to predict scopes for false positive cues. Our end-to-end results would have ranked fourth in the *SEM 2012 shared task with respect to the relevant subtasks.

In the time passed since the shared task, the best published results on the evaluation data appears to be for the system of Packard et al. (2014). Building on top of the UiO_1 system, it obtains a scope-level F1 of 73.1. Depending on the goal, however, F-score in isolation is not the only relevant dimension for system comparisons. The goal of the current work is to create a practically usable tool. For an applied and practical setting, it is also relevant to consider other system properties, like the number of dependencies, platform compatibility, the degree of manual engineering – which can in turn affect how easy it will be to re-train the system on new data or porting the system to cope with other phenomena, the amount of required linguistic pre-processing, and so on. In the system of Packard et al. (2014), the underlying UiO_1 system (Read et al., 2012) is used for cue prediction and as a second source for scope-prediction. While UiO_1 itself is already a highly engineered system – combining manually defined heuristics and statistical ranking of constituent sub-trees – Packard et al. (2014) add a second layer of both (HPSG) parsing and rules (over MRS representations). In sum, the 1.77 point increase in F1 compared to the current system comes at the cost of substantially increased complexity. Importantly though, the full system pipeline is also not publicly available.[3]

For the BiLSTM scope classifier of Fancellu et al. (2016) discussed in Section 4, no results are reported for cue classification, and scope results are only reported for gold cues.[4] Although the code for the BiLSTM scope model is made available, end-to-end results can not be compared without a cue classifier.

[3]The paper of Packard et al. (2014) points to code for replicating the reported experiments, but this only includes support for computing the final layer of 'MRS crawling'. The system of Packard et al. (2014) also relies on cue- and scope predictions from the so-called UiO_1 system of Read et al. (2012), however, and these predictions are only provided in the form of pre-computed system output for the *SEM shared task data; the underlying UiO_1 system is not itself available.

[4]In the system comparison reported by Fancellu et al. (2016), the results of the *SEM shared task competition systems are based on predicted cues while the results of Packard et al. (2014) and Fancellu et al. (2016) are for gold cues, making them not comparable.

6 Future work

One possible improvement of the system would be to extend the scope resolution with post-processing heuristics for targeting discontinuous scopes. The best overall system in the *SEM shared task implemented this (Read et al., 2012), and while the rules themselves require some linguistic understanding, they would be fairly straightforward to implement. There are also certain multi-word cues occurring in the data set that are not covered by the heuristics currently implemented in the system.

Beyond the multi-word cue heuristics, our implementation is abstract in the sense that it is not hard-coded for negation, instead relying on models to be learned automatically from any data using a CoNLL style format similar to that of the *SEM shared task. Importantly, this means that the tool could be trained for other similar tasks, such as speculation detection, as long as cues and scopes are marked. One interesting direction here would be to convert the annotations of the BioScope corpus (Vincze et al., 2008) to the format used by the Conan Doyle corpus. This would allow training of both speculation and negation detection models for biomedical data, and also to test cross-domain effects. Such a conversion is not entirely trivial, however, as the resources differ not merely in terms of format but also the underlying annotation rules. Developing such a mapping could greatly benefit this research field, also making it possible to use data from different domains.

7 Conclusion

This paper has presented an open-source tool for detecting negation cues and their in-sentence scopes. Despite the substantial amount of previous work on negation detection, this has not left much in terms of reusable tools. The presented toolkit mostly relies on machine-learned models, based on a maximum-margin approach. While pre-trained models for English are distributed along with the code, users can also train their own models. In terms of learning frameworks and features, the system design draws on best practice from the existing literature on negation detection, aiming for a simple and portable system that still achieves competitive performance. For future work we plan to also use the tool for training and testing models for speculation detection.

References

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on Multilingual Dependency Parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164, New York, USA.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 173–180, Ann Arbor, MI, USA.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies. A cross-linguistic typology. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 4585–4592, Reykjavik, Iceland.

Federico Fancellu, Adam Lopez, and Bonnie Weber. 2016. Neural Networks for Negation Scope Detection. In *Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics*, pages 495–504, Berlin, Germany.

Richard Farkas, Veronika Vincze, Gyorgy Mora, Janos Csirik, and Gyrgy 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 Natural Language Learning*, pages 1–12, Uppsala, Sweden.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO_2: Sequence-Labeling Negation Using Dependency Features. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 319–327, Montreal, Canada.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, USA.

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 (*SEM)*, pages 265–274, Montreal, Canada. Association for Computational Linguistic.

Roser Morante and Walter Daelemans. 2009. A metalearning approach to processing the scope of negation. In *Proceesings of the Thirteenth Conference on Computational Natural Language Learning*, pages 21–29, Boulder, Colorado, USA.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation in Conan Doyle stories. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 1563–1568, Montreal, Canada.

Andreas C. Müller and Sven Behnke. 2014. PyStruct - Learning Structured Prediction in Python. *Journal of Machine Learning Research*, 15:2055–2060.

Woodley Packard, Emily M. Bender, Jonathon Read, Stephan Oepen, and Rebecca Dridan. 2014. Simple Negation Scope Resolution through Deep Parsing: A Semantic Solution to a Semantic Problem. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 69–78, Baltimore, USA.

Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2012. UiO_1: Constituent-Based Discriminative Ranking for Negation Resolution. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 310–318, Montreal, Canada.

Vladimir N. Vapnik. 1995. The Nature of Statistical Learning Theory. *Springer-Verlag*.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers and the role of syntax. *Computational Linguistics*, 38:369–410.

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

Veronika Vincze, Gyrgy Szarvas, Richrd Farkas, Gyrgy Mra, and Jnos Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(11).

James Paul White. 2012. UWashington: Negation Resolution using Machine Learning Methods. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 335–339, Montreal, Canada.