

Learning Negation Scope from Syntactic Structure

Nick McKenna

School of Informatics
University of Edinburgh
nick.mckenna@ed.ac.uk

Mark Steedman

School of Informatics
University of Edinburgh
steedman@inf.ed.ac.uk

Abstract

We present a semi-supervised model which learns the semantics of negation purely through analysis of syntactic structure. Linguistic theory posits that the semantics of negation can be understood purely syntactically, though recent research relies on combining a variety of features including part-of-speech tags, word embeddings, and semantic representations to achieve high task performance. Our simplified model returns to syntactic theory and achieves state-of-the-art performance on the task of Negation Scope Detection while demonstrating the tight relationship between the syntax and semantics of negation.

1 Introduction

Negation is a semantic phenomenon in natural language which varies significantly. For example, “Sherlock did not solve the case” contains a simple negation cued by “not.” However, there are many cue words such as “without” and “nothing,” and word affixes like “un-” which instantiate negation, and their effect on meaning can be different and dependent on context.

We approach the meaning of negation using a logical semantics, in which natural language negation is expressed with the negation operator on logical expressions (Horn, 1989; Horn and Wansing, 2017), such as in $\neg\text{SOLVE}(\textit{Sherlock}, \textit{case})$. Further, in truth-theoretic logic the meaning of negation is simply the inverted truth value of this expression. Capturing the meaning of a negation cue in language can thus be understood as simply identifying the negated expression. This is the task of Negation Scope Detection (NSD): given a negation cue in a sentence, identify the sentence tokens which make up the negated expression.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

1.1 Negation Scope Detection Task

The task of NSD has been approached in several formulations using different annotation schemes (Kim et al., 2008; Szarvas et al., 2008), but these older tasks do not define negation semantically, in contrast to the commonly accepted *SEM2012 Shared Task competition (*SEM, 2012). Under the *SEM definition cues scope over events and participants, either directly or via predicate arguments and complements. Linguistic theory explains negation semantics in terms of events, which are built from syntactic phrase structures (Huddleston and Pullum, 2005), i.e. that negation doesn’t just scope over individual words, but rather whole phrases and clauses which make up events. The following examples drawn from the *SEM2012 dataset of Conan Doyle writing illustrate how scope is built from phrase structures. Cues are in **bold*** with underlined scope.

(1) Well, sir, I thought **no*** good could come of it.

Example (1) contains the main verb “thought” which takes a complement clause. This clause contains a simple verb “come” negated by its negative subject “no good.” The correct negation scope covers this verb, its arguments, and its modifier “could” (leaving out the cue itself, following *SEM convention). Figure 1 (a) shows that scope corresponds to clause boundaries in the syntactic tree.

(2) He saw him once or twice but he is a deep one and gives **nothing*** away.

The scope in example (2) has discontinuous span. In this sentence “he” is a subject shared by two clauses in coordination, but only one of these is negated. Figure 1 (b) illustrates this. As cases add complexity it’s clear that NSD requires reasoning about the underlying structure of the sentence.

1.2 Contributions

In this work we build on the theory of negation scope as a syntactic phenomenon and reframe the task of NSD as a tree tagging problem over syntactic constituents. We develop a new Structural Tree Recursive Neural Network for this task which labels scope for constituents conditioned on just the syntactic structure of the sentence, with no other features. Our model achieves the highest score to date on the *SEM2012 dataset. We further show that adding word embedding features does not improve results, demonstrating that efficient use of syntax is all you need to perform well on this task.

2 Previous Work

There have been a variety of approaches to NSD. The *SEM shared task dataset is the only resource available with semantic annotations (other NSD datasets are annotated for different goals), but there are many models which use *SEM to compare with (Morante and Blanco, 2012).

Fancellu et al. (2016) show the most recent, best-performing model on *SEM. It is a BiLSTM sequence tagging model which produces in-scope/out-of-scope classifications for each word, and jointly learns word embeddings with additional part-of-speech features for each token. However, further analysis (Fancellu et al., 2017) shows the model is over-reliant on punctuation like commas (“;”) which often mark scope boundaries in English and especially in Conan Doyle’s older style of writing (the content of the *SEM dataset). The model does not learn a very robust semantics without these markers, yet also does not make use of the full syntactic structure of sentences. Fancellu et al. (2018) address this with a Dependency-LSTM model which processes dependency trees using encodings both for words and dependency relations. On a modified *SEM dataset this model slightly improves over the BiLSTM when they are ensembled together, showing again that the BiLSTM method lacks some structural understanding.

The original winner of *SEM, UiO_1 (Read et al., 2012), uses an SVM classifier on constituents that contain the cue. Constituents are shown to be useful, but this is not enough to capture discontinuous scopes (see example (2)) which do not align to a single constituent. Read adds extra heuristics to help with this, and Rosenberg (2013) continues from this with a comprehensive set of syntactic heuristics deployed on dependency graphs. This

model performs well and demonstrates the strength of syntax by itself for capturing negation semantics.

Packard et al. (2014) and Li et al. (2010) approach NSD “in the semantic domain.” Packard first induces a Minimal Recursion Semantics parse of a sentence (Copestake et al., 2005), then crawls it to identify negated predicates and arguments. Though intuitive, inducing a full parse may overshoot the problem — by itself the crawler doesn’t compete because it loses much information during parsing and backtracking to sentence words. Ensembling with UiO_1 provides a boost over UiO_1 itself, which indicates that MRS parsing provides complementary information, possibly due to induction of additional structure.

3 A Structural Approach

We argue for a solution returning to syntax which makes per-word scope judgements conditioned on the full syntactic structure of the sentence. Section 1.1 discusses the theoretical basis in syntax for negation semantics and section 2 details successful models on the task. While syntax is used by several models to a degree, full trees are rarely used and recent models rely on additional features like word embeddings to achieve high performance.

Combinatory Categorical Grammar (CCG) is a nearly context-free constituency grammar capable of describing complex phenomena like coordination (Steedman, 2000). CCG also has transparency between syntax and semantics: a CCG syntactic parse may be transformed into a logical form in terms of events, similar to the Sherlock example in section 1. Steedman (2011) separately formalizes a theory for computing the polarity of lexical items using a CCG-based calculus, which encourages an automated approach to learning negation scope by example.

Figure 1(a) shows the CCG parse tree for example (1). Commonly, simple independent clauses in English form one continuous scope span, which is distinguishable in syntax. The scope of negation in (1) is simply the complement clause.

However, it is not always the case that negation scope aligns cleanly to subtrees, like in example Figure 1(b), the parse tree for sentence (2). CCG provides structural insight which alleviates this problem. In this example CCG’s explicit modeling of coordination makes it straightforward to identify the subject to the left of the coordinated dependent clauses.

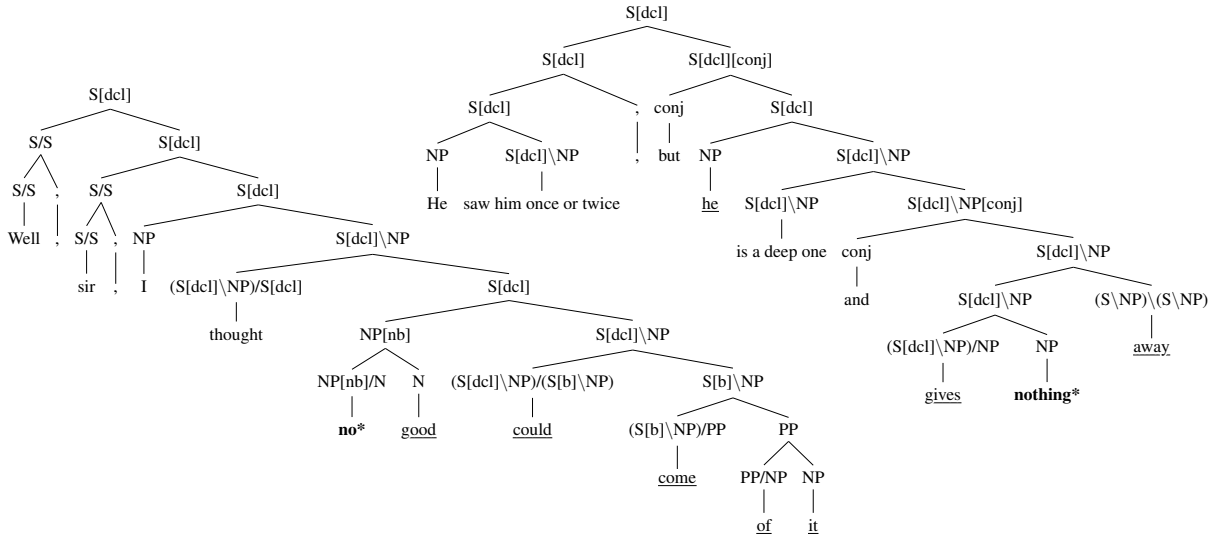


Figure 1: (a) (left) The cue “no” negates the subject “good” and thus scopes over the related event. This corresponds cleanly with the constituent subtree (headed by S[decl]) which spans that text. (b) (right) The cue “nothing” lies in a dependent clause, lacking a subject. This can be found on the left using the CCG parse structure.

Understanding how semantics is built from phrase structures is key to our method. Analyses from additional sentences support this (see appendix for parse trees).

(3) I never* hurt man or woman in my life that I know of.

(4) Some people without* possessing genius have a remarkable power of stimulating it.

4 Methods

The Global Belief Tree Recursive Neural Network of Paulus et al. (2014) is adapted to solve this task. The GB-TRNN takes as input a binarized syntax tree (satisfied by CCG) and starting state vectors for each word in the sentence, referred to now as tree leaves. It first recursively combines constituent states from leaves to root in an upward pass, building up to a single, global state vector. Conditioned on this state, the downward pass recursively unfolds the global vector following the parse tree back down from the root to the leaves. The output for each constituent in the downward pass can be used to produce classifications conditioned on the entire tree.

We add additional syntactic target inputs in both passes and refer to this new architecture as a Structural Tree Recursive Neural Network (STRNN). These additional inputs represent the current kind of CCG combination. In the upward step the parent tag (composed category) is additionally passed in

with the two child states, while in the downward step the two child tags (decomposed categories) are passed in addition to the parent state.

The STRNN learns one embedding matrix for CCG syntactic categories $E_{Cat} \in \mathbb{R}^{|V_{CCG}| \times s}$, where V_{CCG} is the CCG category vocabulary (~ 400 tags) and the embedding size $s = 50$. Figure 2 shows a diagram of the STRNN.

Upward Pass. The initial state \mathbf{u}_i for a leaf constituent i is the result of the transformation matrix $H \in \mathbb{R}^{(s+c) \times h}$. This consumes a category embedding and the cue feature c to produce an initial upward state \mathbf{u}_i of size $h = 200$. c is a binary indicator expressing if the word is a cue. Note that only a word’s CCG category and cue status are shown to the model.

The STRNN first makes an upward pass through the syntax tree. This pass recursively combines two child constituent states \mathbf{u}_{left} and \mathbf{u}_{right} as well as the target parent category embedding $\mathbf{s}_{parent} \in E_{Cat}$ to produce an upward state for the parent constituent \mathbf{u}_{parent} . A weight matrix $W^\uparrow \in \mathbb{R}^{(2h+s) \times h}$ is learned for this operation.

$$\mathbf{x} = [\mathbf{u}_{left} ; \mathbf{u}_{right} ; \mathbf{s}_{parent}]$$

$$\mathbf{u}_{parent} = \tanh(\mathbf{x}W^\uparrow + \mathbf{b}^\uparrow)$$

Downward Pass. The top level hidden state of size h , GLOBAL^\uparrow , is then transformed with the matrix $G \in \mathbb{R}^{h \times h}$ to make the GLOBAL^\downarrow downward state following Paulus. This is recursively unfolded

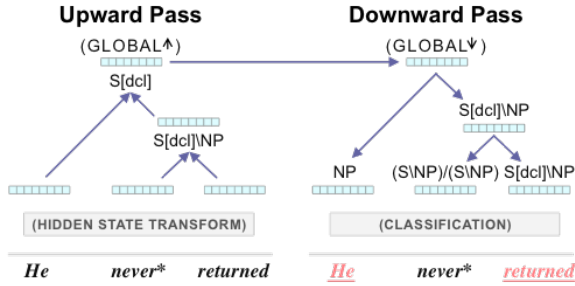


Figure 2: (Left) Upward pass: recursive composition of constituent states from leaves to root. (Right) Downward pass: recursive decomposition of constituent states from root to leaves. **NB:** The model sees only the CCG parse and cue. Word identities are hidden.

down the tree to the leaves using the learned matrix $W^\downarrow \in \mathbb{R}^{(2h+2s) \times 2h}$.

The recursive cell consumes two recurrent inputs following Paulus: the downward parent state \mathbf{d}_{parent} and upward state \mathbf{u}_{parent} . We supply additional inputs which instruct the cell how to decompose the state vector, the target child category embeddings $\mathbf{s}_{left}, \mathbf{s}_{right} \in E_{Cat}$. The cell produces a double-wide state vector which is split into left- and right-child state vectors \mathbf{d}_{left} and \mathbf{d}_{right} . Completing the downward pass results in a globally-informed information state \mathbf{d}_j for each constituent j in the syntax tree.

$$\mathbf{x} = [\mathbf{u}_{parent}; \mathbf{d}_{parent}; \mathbf{s}_{left}; \mathbf{s}_{right}]$$

$$[\mathbf{d}_{left}; \mathbf{d}_{right}] = \tanh(\mathbf{x}W^\downarrow + \mathbf{b}^\downarrow)$$

Classification. Classifications are produced using the matrix $C \in \mathbb{R}^{2h \times 2}$ and a softmax activation. C takes as input, for any tree constituent j , the concatenation of \mathbf{u}_j and \mathbf{d}_j vectors. This can be applied in the same way to tree leaves (words) and to any higher tree constituent to produce a scope judgement.

Optimization. The model is semi-supervised. The *SEM dataset provides supervision signal only for which words are in scope, not general constituents. However, like the GB-TRNN, the STRNN produces classifications for all tree constituents, learning the general pattern from the supervision of words. During training the Adam optimizer was used with cross-entropy loss and 0.001 initial learning rate. Regularization is important with a small training set, and we used dropout following Gal and Ghahramani (2016) with recurrent connections set to 0.2 and others 0.5.

Model	Precision	Recall	F_1
Fancellu BiLSTM	92.62	85.13	88.72
MRS <i>Crawler</i>	85.8	68.4	76.1
<i>UiO</i> ₁	81.99	88.81	85.26
STRNN	94.81	84.04	89.10
STRNN + BERT	89.36	86.54	87.93
STRNN + ST	92.56	83.38	87.73
Punctuation			
Baseline	65.60	89.47	75.70

Table 1: Table of Scope Tokens results for the *SEM2012 Negation Scope Detection test set. “ST” is self-trained embeddings learned jointly with the task. “BERT” is with pretrained embeddings.

5 Results

Table 1 shows performance results for the STRNN and key comparison models on the *SEM2012 test corpus. The conventional *SEM test metric is the F_1 measure of individual sentence tokens predicted in-scope of the cue. Not shown here is Fancellu’s Dependency-LSTM, which published results on a modified version of *SEM. The corpus consists of sentences where each word is annotated with a gold label of in-scope or out-of-scope, and the negation cue. We augment these with CCG parses (Stanojević and Steedman, 2019).

Adapted from Fancellu et al. (2017) is a naïve baseline which predicts scope within the nearest punctuation marks left and right of the cue. It performs fairly well because punctuation indicates and defines grammatical structure. Compared to the STRNN this model overpredicts things like subordinate clauses (e.g. sentence (3)) and underpredicts distant arguments separated by e.g. an appositive.

Our base syntactic model outperforms the comparison models on the task, achieving a new state of the art on this dataset. Notably, it performs at least as well as the word embedding-based model, also the highest-scoring model to date, Fancellu’s BiLSTM. Figure 1 actually shows correct model output on the example sentences, demonstrating the model on a simple case and a complex coordination. Given this, it might be interesting to ablate Fancellu’s BiLSTM to see if it still performs well only using embeddings for POS tags. We note that POS tags contain punctuation markers, and as discussed in section 2 this model has been found to lean heavily on punctuation. The BiLSTM would

be able to take advantage of these markers in the same way as before.

Word embeddings frequently boost performance in language disambiguation tasks by incorporating many kinds of information (Mikolov et al., 2013). We tested augmented STRNN models by adding word embeddings to leaves in addition to syntax embeddings. We obtain an embedding for each word in the sentence using BERT (Devlin et al., 2019) and concatenate this word embedding with our basic syntactic vector, which includes the CCG category embedding and cue status for the given word. We then proceed as before with learning and classification. We also tested a variant with randomly initialized word vectors which are jointly learned with the task. We found that both self-trained word embeddings and pretrained BERT embeddings provided no noticeable benefit to F_1 score on the task.

The model classifies all constituents in the syntax tree. We examined local labeling decisions within trees in the development set and found that in 89.9% of cases where a parent constituent has both children classified in-scope, the model also classifies the parent in-scope. On average, a constituent has a 24.1% chance of in-scope classification. This shows the model’s preference for representing scope in larger phrase structures, aligning with the syntactic theory.

5.1 Error Analysis

Development set results for the base STRNN model were analyzed and 18 difficult sentences were found with accuracy below random chance guessing. The largest group of errors (11 sentences) were likely caused by CCG parsing errors. Some have speech-like text with stuttering, parentheticals, etc, and four have improper attachments within the parse tree affecting the scoped text. A few other sentences have fine CCG parses but are very complex, including one with a 19-word noun phrase. Many of these are related to the style of the genre.

To analyze model robustness, Pearson correlations of accuracy vs several factors were calculated. Negation cues do not always take a scope (such as with interjections) so we measure with accuracy instead of F_1 . No correlation with accuracy was found for sentence length, maximum tree depth, cue depth in the tree, or tree balance.

6 Conclusions

We show the STRNN model effectively predicts negation scope using syntactic parse trees. State-of-the-art performance is achieved on the *SEM2012 Shared Task without identifying individual words or extracting features beyond syntax.

This result reverts to earlier theories about the relationship between syntax and negation semantics. Both word embeddings and semantic reasoning require large resource overhead in model parameters and processing, but efficient use of syntax is all that is needed for high performance on this task.

7 Acknowledgements

This work was supported in part by ERC H2020 Advanced Fellowship GA 742137 SEMANTAX, and an Edinburgh and Huawei Technologies Research Centre award.

References

- Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2-3):281–332.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 495–504.
- Federico Fancellu, Adam Lopez, Bonnie Webber, and Hangfeng He. 2017. Detecting negation scope is easy, except when it isn’t. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 58–63.
- Federico Fancellu, Adam Lopez, and Bonnie L. Webber. 2018. [Neural networks for cross-lingual negation scope detection](#). *CoRR*, abs/1810.02156.
- Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

- Laurence Horn. 1989. *A Natural History of Negation*. University of Chicago Press.
- Laurence R. Horn and Heinrich Wansing. 2017. Negation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, spring 2017 edition. Metaphysics Research Lab, Stanford University.
- Rodney Huddleston and Geoffrey K. Pullum. 2005. *A Student's Introduction to English Grammar*. Cambridge University Press.
- Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10.
- Junhui Li, Guodong Zhou, Hongling Wang, and Qiaoming Zhu. 2010. [Learning the scope of negation via shallow semantic parsing](#). In *Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10*, pages 671–679, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Tomas Mikolov, Scott Wen-tau Yih, and Geoffrey Zweig. 2013. [Linguistic regularities in continuous space word representations](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics.
- Roser Morante and Eduardo Blanco. 2012. *SEM2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 265–274. Association for Computational Linguistics.
- 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 (Volume 1: Long Papers)*, volume 1, pages 69–78.
- Romain Paulus, Richard Socher, and Christopher D Manning. 2014. [Global belief recursive neural networks](#). In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2888–2896. Curran Associates, Inc.
- Jonathon Read, Erik Velldal, Lilja , and Stephan Oepen. 2012. [Uio1: Constituent-based discriminative ranking for negation resolution](#). In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, SemEval '12*, pages 310–318, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Sabine Rosenberg. 2013. Negation triggers and their scope. Master's thesis, Concordia University.
- *SEM. 2012. [*SEM2012: First joint conference on lexical and computational semantics](#).
- Miloš Stanojević and Mark Steedman. 2019. [CCG parsing algorithm with incremental tree rotation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 228–239, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.
- Mark Steedman. 2011. [Negation and polarity](#). In *Taking Scope*, pages 175–208. The MIT Press.
- György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. [The bioscope corpus: Annotation for negation, uncertainty and their scope in biomedical texts](#). In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, BioNLP '08*, pages 38–45, Stroudsburg, PA, USA. Association for Computational Linguistics.