The Case of Imperfect Negation Cues: A Two-Step Approach for Automatic Negation Scope Resolution

Anonymous ACL submission

Abstract

Neural network-based methods are the state of the art in negation scope resolution. However, they often use the unrealistic assumption that cue information is completely accurate. Even if this assumption holds, there remains a dependency on engineered features from state-of-the-art machine learning methods. The current study adopted a two-step negation resolving approach to assess whether a bidirectional long short-term memory-based method can be used for cue detection as well, and how inaccurate cue predictions would affect the scope resolution performance. Results suggest that the scope resolution performance is most robust against inaccurate information for models with a recurrent layer only, compared to extensions with a conditional random field layer or a post-processing algorithm. We advocate for more research into the application of automated deep learning on negation cue detection and the effect of imperfect information on scope resolution.

1 Introduction

Negation is a complex grammatical phenomenon that has received considerable attention in the biomedical Natural Language Processing (BioNLP) domain. Negations play an important role in the semantic representation of biomedical text, because they reverse the truth value of propositions (Morante and Blanco, 2012). Therefore, correct negation handling is a crucial step whenever the goal is to derive factual knowledge from biomedical text.

We can distinguish two ways to approach negations in medical text: negation detection and negation resolving. Negation detection is a form of assertion identification, in this case, determining whether a certain statement is true or false, or whether a medical condition is absent or present (Mutalik et al., 2001; Chapman et al., 2001; Sanchez-Graillet and Poesio, 2007; Huang and Lowe, 2007; Peng et al., 2018; Bhatia et al., 2018; Chen, 2019; Sykes et al., 2021). Negation resolving shifts the focus towards the token level by approaching the problem as a sequence labeling task (Morante et al., 2008). This task is typically divided into two sub tasks: (1) detecting the negation *cue*, a word expressing negation and (2) resolving its *scope*, the elements of the text affected by it. A cue can also be a morpheme ("*im*possible") or a group of words ("not at all"). As an example, in the following sentence the cue is underlined and its scope is enclosed by square brackets:

"I am sure that [<u>neither</u> apples <u>nor</u> bananas are blue]."

Recently, researchers adopted neural network-based approaches to resolve negations (Fancellu et al., 2016, 2017; Lazib et al., 2020). This approach is shown to be highly promising, but most methods solely focus on scope resolution, relying on gold cue annotations. As Read et al. (Read et al., 2012) point out: "It is difficult to compare system performance on sub tasks, as each component will be affected by the performance of the previous." This comparison will not be easier when the performance on a sub task is not affected by the performance of the previous component.

The main advantage of deep learning methods is their independence of manually created features, in contrast to other methods. However, by aiming at scope resolution only, they indirectly still use these features, or assume 100% accurate cues. For complete automatic negation resolving, a neural network model should detect the cue by itself. This raises two questions:

1. How would a neural network-based model perform on the cue detection task?

2. How would a neural network-based model

Table 1: Example of a token sequence and its cue and scope labels.

Tokens	it	had	no	effect	on	IL-10	secretion	.
Cue labels	NC	NC	C	NC	NC	NC	NC	NC
Scope labels	O	O	C	A	A	A	A	O

perform on the scope resolution task with imperfect cue information?

The current study addresses these questions by applying a Bi-directional Long Short-Term Memory (BiLSTM) model (Fancellu et al., 2016) to both stages of the negation resolving task. A BiLSTM model has proven to be good in various NLP tasks, yet not a very complex architecture. We develop the proposed model and their improvements on the BioScope Abstracts and Full Papers sub corpora (Vincze et al., 2008). The results suggest that word embeddings alone can detect cues reasonably well, but there still exist better alternatives for this task. As expected, scope resolution performance suffers from imperfect cue information, but remains acceptable on the Abstracts sub corpus.

As a secondary aim, the current study explores different methods to ensure continuous scope predictions. Since the BioScope corpus only contains continuous scopes, the Percentage Correct Scopes will likely increase after applying such a method. We compare a post-processing algorithm (Morante et al., 2008) with a Conditional Random Field (CRF) layer (Fancellu et al., 2017). The results suggest that both methods are effective, although the post-processing negatively affects the token-based performance.

2 Task Modeling

Let a sentence be represented by a token sequence $\mathbf{t} = (t_1\ t_2\ \cdots\ t_n)$. Following Khandelwal and Sawant (Khandelwal and Sawant, 2020), we use the following labeling scheme for the cue detection task: For $k = 1, \ldots, n$, token t_k token is labeled

- **C** if it is annotated as a single word cue or a discontinuous multiword cue,
- **MC** if it is part of a continuous multiword cue and
- **NC** if it is not annotated as a cue.

The scope label of token t_k token is

- **O** if it is outside of the cue's negation scope,
- **B** if it is inside the negation scope, *before* the first cue token,
- **C** if it is the first cue token in the scope and
- **A** if it is inside the negation scope, *after* the first cue token.

For each sentence, Task 1 is to predict its cue sequence $\mathbf{c} = \{\mathbf{NC}, \mathbf{C}, \mathbf{MC}\}^n$ given its token sequence $\mathbf{t}$ and Task 2 is to subsequently predict the scope sequence $\mathbf{s} = \{\mathbf{O}, \mathbf{B}, \mathbf{C}, \mathbf{A}\}^n$ given $\mathbf{t}$ and $\mathbf{c}$. As an example, the token sequence $\mathbf{t}$ with gold cue and scope labels of "It had [**no** effect on IL-10 secretion]." are given in Table 1.

2.1 Performance measures

To measure performance, we evaluate whether the tokens are correctly predicted as cue or non-cue (Task 1) and as outside or inside the scope (Task 2). At the token level, both tasks are evaluated by precision, recall and F1 measures.

At the scope level, we report the percentage of exact cue matches (PECM) over the number of negation sentences for Task 1. All cue tokens in the sentences have to be correctly labeled to count as an exact match. For Task 2, we adopt the Percentage of Correct Scopes (PCS) as a measure of performance, the percentage of gold negation scopes that are completely match. To evaluate the effectiveness of a 'smoothing' method, we compute the Percentage of Continuous Predictions (PCP) over all scope predictions.[1]

3 Model Architecture

In this section, we describe the proposed model architectures for Task 1 and Task 2. Both tasks are performed by a neural network consisting of an embedding layer, a BiLSTM layer and a softmax layer (Figure 1). For Task 1, we define a baseline model with an embedding layer and a softmax. For both

[1] Let the left and right boundary of a scope be defined as $k_L = \min\{k | s_k \in \{\mathbf{B}, \mathbf{C}, \mathbf{A}\}\}$ and $k_R = \max\{k | s_k \in \{\mathbf{B}, \mathbf{C}, \mathbf{A}\}\}$, respectively. We define a scope to be continuous if $t_k = 1$ for all $k_L \leq k \leq k_R$, and discontinuous otherwise.

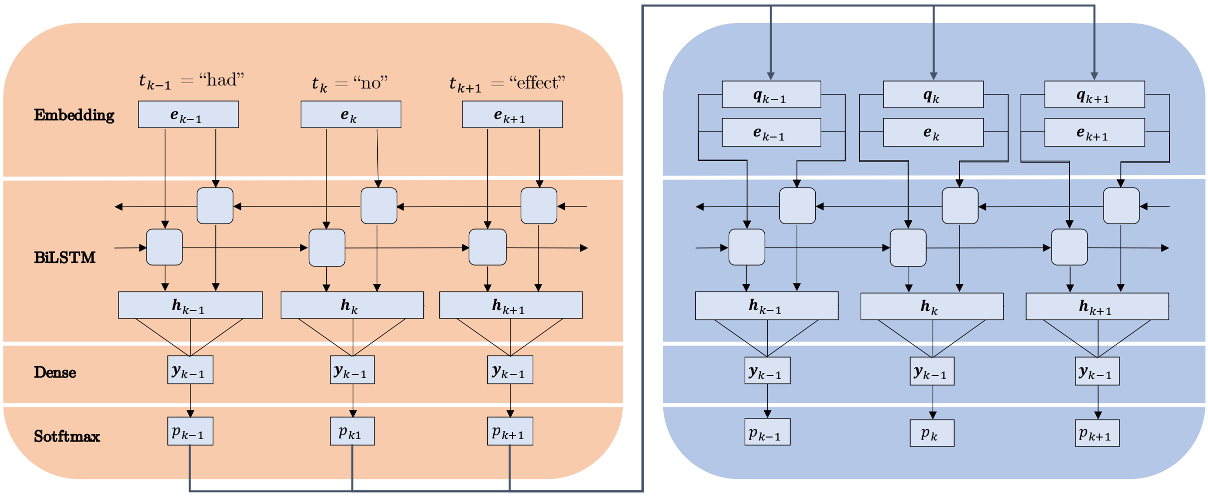

Figure 1: Schematic representation of the BiLSTM model for cue detection (left) and scope resolution (right), for the example sentence "It had no effect on IL-10 secretion." at $k = 3$.

tasks, we add a model where the softmax layer is replaced by a CRF layer to obtain a joint prediction for the token sequence. Finally, we discuss how the models were trained.

3.1 Word Embeddings for cue detection

The token sequence $\mathbf{t} = (t_1 \ \cdots \ t_n)$ is the only input for the cue detection models. Let $E^{d \times v}$ be an embedding matrix, where d is the embedding dimension and v is the vocabulary size. Then, each token in $\mathbf{t} = (t_1 \ \cdots \ t_n)$ is represented by a pre-trained BioWordVec (Chen et al., 2019) embedding $\mathbf{e} \in \mathbb{R}^d$ corresponding to its vocabulary index. These embeddings were trained by the Fasttext subword embedding model with a context window size of 20 (Bojanowski et al., 2017) on the MIMIC-III corpus (Johnson et al., 2016). This model is able to include domain-specific subword information into its vector representations. Out-of-vocabulary (OOV) tokens were represented by a d-dimensional zero vector.

Word embeddings may represent features that are already informative enough for the cue detection task. Therefore, we define a baseline model where the embeddings are directly passed to a 3-unit dense layer with weights $W_s^{3 \times d}$ and bias $\mathbf{b}_s \in \mathbb{R}^3$. The output vector

$$\mathbf{y}_k = W_s \mathbf{e}_k + \mathbf{b}_s = (y_k^{NC}, y_k^{C}, y_k^{MC})$$

contains to the 'confidence' scores of tagging token k as a non-cue, cue or multiword cue, respectively. These scores are used to obtain the final prediction label $p_k = \mathrm{softmax}(\mathbf{y}_k)$, where the softmax function $\mathbb{R}^3 \to \{\mathbf{NC}, \mathbf{C}, \mathbf{MC}\}$ is given by

$$\mathbf{y} \mapsto \left\{ \frac{e^{y^{NC}}}{Z}, \frac{e^{y^{C}}}{Z}, \frac{e^{y^{MC}}}{Z} \right\}, \quad Z = \sum_{y \in \mathbf{y}} e^y.$$

3.2 BiLSTM for cue detection

In the BiLSTM model, the token embeddings $(\mathbf{e}_1 \ \cdots \ \mathbf{e}_n)$ are passed to a BiLSTM layer (Graves and Schmidhuber, 2005) with $2U$ units, U in the forward direction and U in the backward direction. We represent an LSTM layer as a sequence of n identical cells. A cell at token k is described by the following set of equations corresponding to its input gate $\mathbf{i}_k$, forget gate $\mathbf{f}_k$, output gate $\mathbf{o}_k$, candidate memory state $\tilde{\boldsymbol{\gamma}}_k$, memory state $\boldsymbol{\gamma}_k$ and hidden state $\mathbf{h}_k$, respectively:

$$\begin{aligned}
\mathbf{i}_k &= \sigma\big(W_e^{(i)}\mathbf{e}_k + W_h^{(i)}\mathbf{h}_{k-1} + \mathbf{b}^{(i)}\big),\\
\mathbf{f}_k &= \sigma\big(W_e^{(f)}\mathbf{e}_k + W_h^{(f)}\mathbf{h}_{k-1} + \mathbf{b}^{(f)}\big),\\
\mathbf{o}_k &= \sigma\big(W_e^{(o)}\mathbf{e}_k + W_h^{(o)}\mathbf{h}_{k-1} + \mathbf{b}^{(o)}\big),\\
\tilde{\boldsymbol{\gamma}}_k &= \tanh\big(W_e^{(\tilde{\gamma})}\mathbf{e}_k + W_h^{(\tilde{\gamma})}\mathbf{h}_{k-1} + \mathbf{b}^{(\tilde{\gamma})}\big),\\
\boldsymbol{\gamma}_k &= \mathbf{f}_k \odot \boldsymbol{\gamma}_{k-1} + \mathbf{i}_k \odot \tilde{\boldsymbol{\gamma}}_k,\\
\mathbf{h}_k &= \mathbf{o}_k \odot \tanh(\boldsymbol{\gamma}_k),
\end{aligned}$$

where $W_e^{U \times d}$ denote the weight matrices for the token embeddings, $W_h^{U \times U}$ denotes the recurrent weight matrix, $\mathbf{b} \in \mathbb{R}^u$ is a bias vector, $\odot$ denotes the Hadamard product, σ denotes the sigmoid function[2] and tanh denotes the hyperbolic tangent function.[3] The hidden state of the forward layer and

[2] The function $\mathbb{R} \to (0, 1)$ given by $x \mapsto 1/(1 + e^{-x})$

[3] The function $\mathbb{R} \to (-1, 1)$ given by $x \mapsto (e^x - e^{-x})/(e^x + e^{-x})$

backward layer are concatenated to yield a representation $\overleftrightarrow{\mathbf{h}}_k = (\overrightarrow{\mathbf{h}}_k; \overleftarrow{\mathbf{h}}_k) \in \mathbb{R}^{2u}$ for token k. For each token, the output $\overleftrightarrow{\mathbf{h}}_k$ of the BiLSTM layer is fed into a 3-unit softmax layer with weights $W_s^{3\times 2U}$ and bias $\mathbf{b}_s \in \mathbb{R}^3$, as defined in the baseline model.

3.3 Adding a conditional random field layer

Although the context around token t is captured by the LSTM cell, the model will still assume independence between the token predictions when it maximizes a likelihood function. Alternatively, we can replace the softmax layer of the cue detection models by a Conditional Random Field (CRF) layer (Lafferty et al., 2001) to create a dependency between the predictions of adjacent tokens. This allows the model to learn that a single cue token is surrounded by non-cue tokens, and that a multi-word cue token is always followed by a next one.

Let $Y = (\mathbf{y}_1 \ \cdots \mathbf{y}_n)$ be the $3 \times n$ matrix of model predicted scores

$$\begin{pmatrix} y_1^{NC} & y_2^{NC} & \cdots & y_n^{NC} \\ y_1^{C} & y_2^{C} & \cdots & y_n^{C} \\ y_1^{MC} & y_2^{MC} & \cdots & y_n^{MC} \end{pmatrix}.$$

Consider all possible label sequences enclosed by start/end labels $\mathcal{P} = \{\text{start}\} \times \{\mathbf{NC}, \mathbf{C}, \mathbf{MC}\}^n \times \{\text{end}\}$. Let $\mathbf{p}^* \in \mathcal{P}$ and let $T \in \mathbb{R}^{5\times 5}$ be a matrix of transition scores, such that score $T_{i,j}$ corresponds to moving from the i-th to the j-th label in the set $\{\mathbf{NC}, \mathbf{C}, \mathbf{MC}, \text{start}, \text{end}\}$. Then, a linear CRF yields a joint prediction for a token sequence $\mathbf{t}$ by attaching it a global score

$$S(\mathbf{t}, \mathbf{c}, \mathbf{p}^*) = \sum_{k=1}^{n} Y_{p_k^*, k} + \sum_{k=0}^{n} T_{p_k^*, p_{k+1}^*}.$$

The model predicts the label sequence with the maximum score among all possible label sequences:

$$\mathbf{p} =_{\mathbf{p}^* \in \mathcal{P}} S(\mathbf{t}, \mathbf{c}, \mathbf{p}^*)$$

3.4 BiLSTM for scope resolution

The scope resolution model accepts as input the token sequence $\mathbf{t}$ and a cue vector $(c_1 \ \cdots \ c_n) \in \{0,1\}^n$, where $c_k = 0$ if the (gold or predicted) cue label of token k is $\mathbf{NC}$ and $c_k = 1$ otherwise. The embedding layer yields a cue embedding $\mathbf{q} \in \{1\}^d$ if $c_k = 1$ and $\mathbf{q} \in \{0\}^d$ if $c_k = 0$. For the token input, we use the same embedding matrix $E^{v\times d}$ as in the previous model.

The token and cue embeddings are passed to a BiLSTM layer with $2U$ units. An LSTM layer is well-suited for the scope resolution, since it can capture long term dependencies between a cue token and a scope token. The bidirectionality accounts for the fact that a scope token can be located to the left and the right of a cue token. The hidden state of the forward layer and backward layer are concatenated to yield a representation $\overleftrightarrow{\mathbf{h}}_k = (\overrightarrow{\mathbf{h}}_k; \overleftarrow{\mathbf{h}}_k) \in \mathbb{R}^{2u}$ for token k.

For each token, the output $\overleftrightarrow{\mathbf{h}}_k$ of the BiLSTM layer is fed into a 4-unit dense layer with weights $W_s^{2\times 2U}$ and bias $\mathbf{b}_s \in \mathbb{R}^2$. The output vector

$$\mathbf{y}_k = W_s \overleftrightarrow{\mathbf{h}}_k + \mathbf{b}_s = (y_k^O, y_k^B, y_k^C, y_k^A)$$

contains to the 'confidence' scores of the possible scope labels. These scores are used to obtain the final prediction label $p_k = \text{softmax}(\mathbf{y}_k)$.

3.5 BiLSTM + CRF for scope resolution

A BiLSTM+CRF model is also used for the scope resolution task. The model might learn that certain sequences are impossible, for example, that a **B** will never follow a **C**. Moreover, we expect that the model will yield more continuous scope predictions.

3.6 Model training

The objective of the models is to maximize the likelihood $\mathcal{L}(\boldsymbol{\Theta})$ of the correct predictions $\mathbf{p}$ compared to the gold labels $\mathbf{g} = (g_1 \ \cdots \ g_n)$, with $\boldsymbol{\Theta}$ the set of trainable model parameters and $\mathbf{X}$ the inputs of the model. For the BiLSTM models, this likelihood is

$$\mathcal{L}(\boldsymbol{\Theta}) = \prod_{k=1}^{n} \left(p_k(\boldsymbol{\Theta}, \mathbf{X})\right)^{g_t} \left(1 - p_k(\boldsymbol{\Theta}, \mathbf{X})\right)^{1-g_t},$$

for the BiLSTM-CRF models, this likelihood is

$$\mathcal{L}(\boldsymbol{\Theta}) = \frac{e^{S(\mathbf{X}, \mathbf{p})}}{\sum_{\mathbf{p}^* \in \mathcal{P}} e^{S(\mathbf{X}, \mathbf{p}^*)}}.$$

Hyperparameters The models were compiled and fitted with the Keras functional API for TensorFlow 2.3.1 in Python 3.7.6 (Abadi et al., 2016; Van Rossum et al., 2000). Based on validation results, we selected the Adam optimizer with an initial learning rate 0.001 with step decay to find optimal values for $\boldsymbol{\Theta}$. Scope resolution models were trained on 30 epochs with a batch size of 32. The

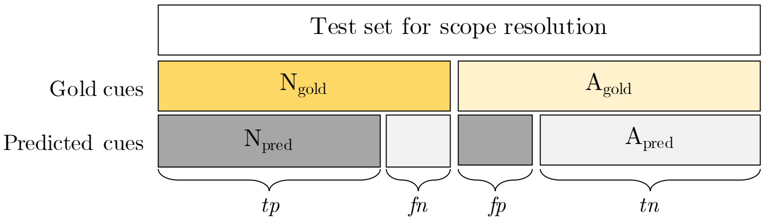

Figure 2: Visualization of negation sentences (N) and assertion sentences (A) in the test set, under different circumstances. Note: tp=true positives, fn=false negatives, fp=false positives, tn=true negatives.

cue detection models were trained with early stopping, since the model showed large overfitting on 30 epochs. For the architecture hyperparameters, we selected embedding dimension $d = 200$ and number of units in the LSTM-layer $U = 200$. Embeddings were not updated during training, except for the cue detection baseline model.

3.7 Post-processing

In Task 2, we apply a post-processing algorithm on the predictions of the BiLSTM model to obtain continuous scope predictions (Morante et al., 2008). We first ensure that the cue tokens are labeled as a scope token. In case of a discontinuous negation cue, the tokens between the cue tokens are also labeled as a scope token. The algorithm locates the continuous prediction 'block' containing the cue token and decides whether to connect separated blocks around it, based on their lengths and the gap length between them.

4 Experiments

4.1 Corpus

The current study made use of the Abstracts and Full papers sub corpora from the open access BioScope corpus (Vincze et al., 2008). Together, these sub corpora contain 14,462 sentences. For each sentence, the negation cue and its scope are annotated such that the negation cue is as small as possible, the negation scope is as wide as possible and the negation cue is always part of the scope. Resulting from this strategy, every negation cue has a scope and all scopes are continuous.

One sentence contained two negation instances. We represented this sentence twice, such each copy corresponded to a different negation instance. This resulted in 2,094 (14.48%) negation instances. A description of the sub corpora is provided in Table 2.

Tokenization Biomedical text data poses additional challenges to the problem of tokenization (Díaz and López, 2015). DNA sequences, chemical substances and mathematical formula's appear frequently in this domain, but are not easily captured by simple tokenizers. Examples are "E2F-1/DP1" and "CD4(+)". In the current pipeline, the standard NLTK-tokenizer was used (Loper and Bird, 2002), in accordance with the tokenizer used by the BioWordVec model. This resulted in a vocabulary of 17,800 tokens, with each token present in both sub corpora. Tokenized sentences were truncated (23 sentences) or post-padded to match a length of 100 tokens.

4.2 Experimental set-up

For the experiments, we apply a 70-15-15 train-validation-test split to the sub corpora. First, we train and test the cue detection models. The set of sentences with at least one predicted cue label are passed to Task 2. We use the predicted cue labels of the best model, based on the validation F1. This predicted Negation set consists of true positives and false positives: $N_{\text{pred}} = tp \cup fp$. We define its complement, the predicted Assertion set, as $A_{\text{pred}} = \text{fn} \cup \text{tn}$ and predict an empty negation scope $\mathbf{p} \in \{\mathbf{O}\}^n$ for this set.

The models in Task 2 could be tested on N_{pred}, with predicted cue inputs. However, the model performance will be affected by the presence of false positives and absence of false negatives from Task 1 in this set. To compare this with testing on $N_{\text{gold}} = tp \cup fn$ with gold cue inputs, we need to base our results on the same data. Therefore, we use $N_{\text{gold}} \cup N_{\text{pred}} = tp \cup fn \cup fp$ as a general test set for Task 2, see Figure 2. Note that tn is not needed, since true negatives are not involved in the performance measures.

Table 2: Descriptive statistics of the sub corpora.

	Statistic	Abstracts	Full Papers
Total	Documents	1,273	9
	Sentences	11,994	2,469
	Negation instances	14.3%	15.2%
	Tokens	317,317	69,367
	OOV	0.1%	1.4%
Sentence length n	$n \leq 25$	53.5%	50.6%
	$25 < n \leq 50$	43.2%	42.7%
	$50 < n \leq 75$	3.0%	5.6%
	$75 < n$	0.3%	1.1%
Scope length S	$S \leq 10$	69.9%	72.0%
	$10 < S \leq 30$	24.2%	22.1%
	$30 < S$	58.7%	58.7%
	Avg. S/n	0.33	0.30
Scope bounds	Avg. k_L	16.4	16.2
	Avg. k_R	23.1	22.8
	Avg. k_L/n	0.51	0.47
	Avg. k_R/n	0.76	0.70
	Scope starts with cue	85.5%	78.7%

Note: OOV = Out Of Vocabulary tokens, that is, not appearing in the BioWordVec pre-trained embeddings. Avg. = average.

5 Results and Discussion

5.1 Task 1 performance

The results indicate that BiLSTM-based models can detect negation cues reasonably well in the Abstracts corpus, but perform poorly on the Full Papers corpus. The difference not surprising, since we know from previous studies that most models perform worse on the Full Papers corpus. In Table 3, we report the performance of the proposed methods compared to the current state-of-the-art machine learning and neural network methods. It is clear that the models underperform on both corpora by a large margin.

The most surprising result is that none of the models perform remarkably better than the baseline model of non-trainable word embeddings. Adding a BiLSTM layer even leads to worse performance: The precision and recall measures indicate that less tokens are labeled as a cue with a BiLSTM layer, reducing the false positives, but increasing the false negatives. Apparently, the BiLSTM layer cannot capture more syntactical information needed for cue detection than already present in the embeddings. The embeddings do not benefit from a CRF layer either. It is only with a BiLSTM-CRF combination that the overall performance improves by predicting more non-cue labels for tokens that are indeed not a cue token. Among the currently proposed models, we conclude that the BiLSTM+CRF model is the best for the Abstracts corpus.

In contrast, training the embeddings does lead to a better performance on the Full Papers corpus. Here, the performance measures are more conclusive. The F1 measure is halved after adding a BiLSTM layer to the embeddings, and adding a CRF leads to no predicted cue labels at all. We therefore use the trained embeddings model to obtain the cue predictions for the Full Papers corpus.

5.2 Task 2 performance

Overall, it is clear that the models suffer from imperfect cue information. The F1 on the scope resolution task can decrease up to 9% on the Abstracts corpus and 18% on the Full Papers corpus, when moving from gold to predicted information, see Table 4. The BiLSTM model seems to be the most robust against this effect. The transition scores of a CRF layer might make the model more receptive to cue inputs. When the model is presented a false positive cue, the transition score from an **O**-label to a **C** makes it easier to predict a false positive **C**. It is also clear why the post-processing algorithm performs worse with imperfect cue information, as it guarantees that all false positive cues will receive a false positive scope label. This is confirmed by the sharp drop in precision (14%) and the small drop in recall (4%), see Table 5.

As a secondary aim, we investigated the effect of the CRF layer and the post-processing algorithm on the Percentage of Correct Scopes. In all cases, we see that the post-processing algorithm yields the highest PCS. However, this comes at the cost of a lower F1 measure at the token level when the model receives predicted cue inputs. Another disadvantage of this approach is that is not easily

Table 3: Performance of the cue detection models.

BioScope Abstracts				
Method	P	R	F1	PECM
Baseline	80.59	87.81	84.05	76.95
Emb. train (E)	79.87	89.61	84.46	74.22
E + BiLSTM	84.87	82.44	83.64	78.52
E + CRF	82.62	83.51	83.07	76.95
E + BiLSTM + CRF	83.22	87.10	85.11	80.86
Metalearner (Morante and Daelemans, 2009)	**100**	**98.75**	**99.37**	**98.68**
NegBERT (Khandelwal and Sawant, 2020)	NR	NR	95.65	NR
BioScope Full Papers				
Method	P	R	F1	PECM
Baseline	64.18	62.32	63.24	47.46
Emb. train (E)	60.23	76.81	67.52	49.15
E + BiLSTM	58.33	20.28	30.11	18.64
E + CRF	NaN	0	NaN	0
E + BiLSTM + CRF	60.53	66.67	63.45	45.76
Metalearner (Morante and Daelemans, 2009)	**100**	**95.72**	**96.08**	**92.15**
NegBERT (Khandelwal and Sawant, 2020)	NR	NR	90.23	NR

Note: PECM=Percentage Exact Cue Matches.

Table 4: F1 scores on the scope resolution task with Gold versus Predicted cue inputs.

Abstracts, Cue detection F1 = 85.11			
Method	Gold input	Predicted input	Difference
BiLSTM	90.25	83.90	**6.35**
BiLSTM+CRF	91.58	84.43	7.15
BiLSTM+post	90.17	80.87	9.30
Full Papers, Cue detection F1 = 67.52			
Method	Gold input	Predicted input	Difference
BiLSTM	72.80	56.98	**15.82**
BiLSTM+CRF	76.10	59.19	16.91
BiLSTM+post	73.29	54.79	18.50

transferable to genres where the annotation style is different. For example, discontinuous scopes are quite common in the Conan Doyle corpus (Morante and Daelemans, 2012).

The results indicate that the BiLSTM+CRF model often resolves more scopes completely than the BiLSTM model. This could be partly explained by the increase in continuous predictions, as earlier suggested by Fancellu et al. (Fancellu et al., 2017). However, on the Full Papers corpus with predicted inputs, the CRF-based model yields a lower PCS. The precision and recall measures indicate that the BiLSTM+CRF model predicts more positive cue labels, which may result in scopes that are too wide. We also see that there remains a substantive percentage of discontinuous predictions. This may be solved by higher-order CRF layers, that is, including transitions of label k to label $k + 2$.

6 Conclusion and Future Work

The current study adopted a neural network-based approach to both sub tasks of negation resolving: cue detection and scope resolution. In this way, the task would be completely independent of hand-crafted features, and would more realistically demonstrate the performance on the scope detection task. The study showed that the applicability of the BiLSTM approach does not extend to cue detection: isolated word embeddings are just as effective. These embeddings could capture features that are informative for cue detection, but they need more 'flexible' contextual information to distinguish negative or neutral use of a potential cue token within a given sentence. There are various architectures available that could tackle this problem more effectively: Encoder-Decoder LSTMs (Wang et al., 2016), attention based architectures (Chen, 2019; Khandelwal and Sawant, 2020; Britto and Khandelwal, 2020), hierarchical LSTMs and Embeddings from Language Models (ELMo and BERT, (Peters et al., 2018; Devlin et al., 2018)).

The scope resolution performance of a BiLSTM+CRF-based method with inaccurate cue labels is hopeful. The model still outperforms most early methods, and performs on par with some recent methods. It would be interesting to assess

Table 5: Performance of the scope resolution model on the Abstracts corpus.

BioScope Abstracts							
Cues	Method	P	R	F1	PCS	PCP
Gold	BiLSTM	89.80	90.70	90.25	68.34	87.89
	BiLSTM+CRF	91.07	**92.10**	91.58	70.31	92.19
	BiLSTM+post	90.43	89.92	90.17	72.66	100
	Metalearner (Morante and Daelemans, 2009)	90.68	90.68	90.67	73.36	100
	RecurCRFs* (Fei et al., 2020)	**94.9**	90.1	93.6	**92.3**	-
	NegBERT (Khandelwal and Sawant, 2020)	NR	NR	**95.68**	NR	NR
Pred	BiLSTM	**81.83**	86.08	83.90	58.59	83.07
	BiLSTM+CRF	81.29	**87.82**	**84.43**	58.98	87.40
	BiLSTM+post	76.40	85.90	80.87	60.55	100
	Metalearner (Morante and Daelemans, 2009)	81.76	83.45	82.60	**66.07**	100
BioScope Full Papers						
Cues	Method	P	R	F1	PCS	PCP
Gold	BiLSTM	94.21	59.31	72.80	28.81	88.14
	BiLSTM+CRF	80.87	71.86	76.10	32.20	89.83
	BiLSTM+post	**94.86**	59.72	73.29	32.20	100
	Metalearner (Morante and Daelemans, 2009)	84.47	**84.95**	84.71	**50.26**	100
	NegBERT (Khandelwal and Sawant, 2020)	NR	NR	**87.35**	NR	NR
Pred	BiLSTM	67.69	49.19	56.98	18.64	56.92
	BiLSTM+CRF	57.55	60.93	59.19	16.95	63.08
	BiLSTM+post	49.92	60.73	54.79	22.03	100
	Metalearner (Morante and Daelemans, 2009)	**72.21**	**69.72**	**70.94**	**41.00**	100

Note: PCS = Percentage Correct Scopes, PCP=Percentage Continuous scope Predictions. *These results were reported for the complete BioScope corpus.

the robustness of other neural network-based models against imperfect cue inputs, possibly with different levels and forms of cue accuracy. Additionally, this robustness could be integrated in the approach. For example, we could capture the prediction uncertainty of the cue inputs by feeding the probabilities instead of the labels to the scope resolution model.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Amjad Abu-Jbara and Dragomir Radev. 2012. Umichigan: A conditional random field model for resolving the scope of negation. In ** SEM 2012: 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 2012)*, pages 328–334.

Shashank Agarwal and Hong Yu. 2010. Biomedical negation scope detection with conditional random fields. *Journal of the American medical informatics association*, 17(6):696–701.

Jorge Carrillo de Albornoz, Laura Plaza, Alberto Díaz, and Miguel Ballesteros. 2012. Ucm-i: A rule-based syntactic approach for resolving the scope of negation. In ** SEM 2012: 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 2012)*, pages 282–287.

Miguel Ballesteros, Alberto Díaz, Virginia Francisco, Pablo Gervás, Jorge Carrillo De Albornoz, and Laura Plaza. 2012. Ucm-2: a rule-based approach to infer the scope of negation via dependency parsing. In ** SEM 2012: 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 2012)*, pages 288–293.

Valerio Basile, Bos Johan, Evang Kilian, and Venhuizen Noortje. 2012. Ugroningen: Negation detection with discourse representation structures. In *First Joint Conference on Lexical and Computational Semantics (* SEM)*, pages 301–309. Association for Computational Linguistics.

Parminder Bhatia, Busra Celikkaya, and Mohammed Khalilia. 2018. Joint entity extraction and assertion detection for clinical text. *arXiv preprint arXiv:1812.05270*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Benita Kathleen Britto and Aditya Khandelwal. 2020.

Resolving the scope of speculation and negation using transformer-based architectures. *arXiv preprint arXiv:2001.02885*.

Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

Long Chen. 2019. Attention-based deep learning system for negation and assertion detection in clinical notes. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 10(1).

Qingyu Chen, Yifan Peng, and Zhiyong Lu. 2019. Biosentvec: creating sentence embeddings for biomedical texts. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–5. IEEE.

Noa P Cruz, Maite Taboada, and Ruslan Mitkov. 2016. A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology*, 67(9):2118–2136.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Noa P Cruz Díaz and Manuel J Maña López. 2015. An analysis of biomedical tokenization: problems and strategies. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 40–49.

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)*, 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*, pages 58–63.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Negation and speculation scope detection using recursive neural conditional random fields. *Neurocomputing*, 374:22–29.

Dipesh Gautam, Nabin Maharjan, Rajendra Banjade, Lasang Jimba Tamang, and Vasile Rus. 2018. Long short term memory based models for negation handling in tutorial dialogues. In *The Thirty-First International Flairs Conference*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Binod Gyawali and Thamar Solorio. 2012. Uabcoral: a preliminary study for resolving the scope of negation. In ** SEM 2012: 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 2012)*, pages 275–281.

Yang Huang and Henry J Lowe. 2007. A novel hybrid approach to automated negation detection in clinical radiology reports. *Journal of the American medical informatics association*, 14(3):304–311.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Aditya Khandelwal and Suraj Sawant. 2020. Negbert: A transfer learning approach for negation detection and scope resolution. *arXiv preprint arXiv:1911.04211*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. Uio 2: sequence-labeling negation using dependency features. In ** SEM 2012: 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 2012)*, pages 319–327.

Lydia Lazib, Bing Qin, Yanyan Zhao, Weinan Zhang, and Ting Liu. 2020. A syntactic path-based hybrid neural network for negation scope detection. *Frontiers of computer science*, 14(1):84–94.

Lydia Lazib, Yanyan Zhao, Bing Qin, and Ting Liu. 2019. Negation scope detection with recurrent neural networks models in review texts. *International Journal of High Performance Computing and Networking*, 13(2):211–221.

Hao Li and Wei Lu. 2018. Learning with structured representations for negation scope extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 533–539.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70.

Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M Roch, Heidi Schmidt, Joe Kesterson, Chris Beesley, Paul Dexter, C Max Schmidt, Hongfang Liu, et al. 2015. Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of biomedical informatics*, 54:213–219.

Roser Morante and Eduardo Blanco. 2012. * sem 2012 shared task: Resolving the scope and focus of negation. In * *SEM 2012: 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 2012)*, pages 265–274.

Roser Morante and Walter Daelemans. 2009. A metalearning approach to processing the scope of negation. In *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*, pages 21–29.

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*, pages 1563–1568.

Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the scope of negation in biomedical texts. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 715–724.

Pradeep G Mutalik, Aniruddha Deshpande, and Prakash M Nadkarni. 2001. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the umls. *Journal of the American Medical Informatics Association*, 8(6):598–609.

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)*, pages 69–78.

Yifan Peng, Xiaosong Wang, Le Lu, Mohammadhadi Bagheri, Ronald Summers, and Zhiyong Lu. 2018. Negbio: a high-performance tool for negation and uncertainty detection in radiology reports. *AMIA Summits on Translational Science Proceedings*, 2018:188.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Zhong Qian, Peifeng Li, Qiaoming Zhu, Guodong Zhou, Zhunchen Luo, and Wei Luo. 2016. Speculation and negation scope detection via convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 815–825.

Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2012. Uio1: Constituent-based discriminative ranking for negation resolution. In * *SEM 2012: 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 2012)*, pages 310–318.

Olivia Sanchez-Graillet and Massimo Poesio. 2007. Negation of protein–protein interactions: analysis and extraction. *Bioinformatics*, 23(13):i424–i432.

Elena Sergeeva, Henghui Zhu, Peter Prinsen, and Amir Tahmasebi. 2019. Negation scope detection in clinical notes and scientific abstracts: a feature-enriched lstm-based approach. *AMIA Summits on Translational Science Proceedings*, 2019:212.

Dominic Sykes, Andreas Grivas, Claire Grover, Richard Tobin, Cathie Sudlow, William Whiteley, Andrew McIntosh, Heather Whalley, and Beatrice Alex. 2021. Comparison of rule-based and neural network models for negation detection in radiology reports. *Natural Language Engineering*, 27(2):203–224.

Stuart J Taylor and Sanda M Harabagiu. 2018. The role of a deep-learning method for negation detection in patient cohort identification from electroencephalography reports. In *AMIA Annual Symposium Proceedings*, volume 2018, page 1018. American Medical Informatics Association.

Guido Van Rossum, Fred L Drake, et al. 2000. *Python reference manual*. iUniverse Indiana.

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(11):1–9.

Tong Wang, Ping Chen, Kevin Amaral, and Jipeng Qiang. 2016. An experimental study of lstm encoder-decoder model for text simplification. *arXiv preprint arXiv:1609.03663*.

James Paul White. 2012. Uwashington: Negation resolution using machine learning methods. In * *SEM 2012: 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 2012)*, pages 335–339.

A Appendices

A.1 Related Work

Negation resolving has been tackled by a range of approaches: rule-based methods, Machine Learning (ML) and Conditional Random Fields (CRFs). In this section, we will briefly discuss these approaches, followed by a discussion of neural network-based studies. An brief overview of the performance of earlier proposed methods is provided in Table 6.

Rule-based methods were the first methods used for negation detection, but only later they were applied to negation resolving. Examples of rule-based approaches are the use of regular expression algorithms (Chapman et al., 2001; Mehrabi et al., 2015), pre-defined lexicons and syntax trees, (de Albornoz et al., 2012; Ballesteros et al., 2012) and text representations with formal semantic structures (Basile et al., 2012). Within this approach, it is common to first detect the negation cues, and subsequently resolve their scope.

Although rule-based methods show acceptable performance on both tasks, they do not easily generalize to other domains or even data sets. Machine Learning (ML) classifiers were introduced to overcome this problem, performing on par with or better than rule-based methods (Lapponi et al., 2012; Cruz et al., 2016). Examples are memory-based learning algorithms (Morante et al., 2008), Support Vector Machines (SVM) (Gyawali and Solorio, 2012), metalearning approaches (Morante and Daelemans, 2009) and hybrid methods, combining SVM classifiers with heuristic rules (Read et al., 2012; Packard et al., 2014). Most ML methods are also designed for a two-step procedure where scope resolution is influenced by the accuracy of the cue predictions. Morante et al. (Morante and Daelemans, 2009) showed the importance of this problem by comparing their system with perfect and imperfect cue information, and reported a 8% decrease in token-based F1 measure. Packard et al. (Packard et al., 2014) made a similar comparison and reported a 4% F1 decrease when moving from gold cue annotations to predicted cue labels.

The two-step procedure was also adopted by researchers using Conditional Random Fields (CRF) models. These models are well suited for sequence labeling tasks, since a token sequence can be easily represented as a linear graph. Most of these models achieve acceptable performance on the scope resolution task with the use of predicted cue features and other syntactic features (Agarwal and Yu, 2010; Abu-Jbara and Radev, 2012; White, 2012; Li and Lu, 2018).

Recently, researchers started to investigate the application of neural network models to scope resolution. In this way, hand-crafted features needed for Machine Learning could be replaced by unsupervised features. For example, Qian et al. (Qian et al., 2016) used Convolutional Neural Networks (CNNs) to extract path features and combined these with position features. BiLSTM-based models became the state of the art (Fancellu et al., 2016, 2017; Lazib et al., 2019), capable of integrating word and cue embeddings into their memory cells. Later, Fei et al. (Fei et al., 2020) outperformed this method with a Recursive Neural Network that automatically learns syntactic features, combined with a CRF layer. All these methods aim at the scope resolution task, assuming gold cue information.

More recently, transformer-based models have shown to be the current state of the art (Khandelwal and Sawant, 2020; Britto and Khandelwal, 2020). Importantly, these models are also capable of detecting negation cues. In the second stage of the task, they use a method that replaces the original token in the sentence by a special cue token. Currently, this stage is only performed with gold cue tokens.

The tasks can also be solved separately, that is, by not passing information of the first sub task to the second. Gautam et al. (Gautam et al., 2018) developed an Encoder-Decoder LSTM for this approach. They showed that this model can detect negation cues with a 100% precision in conversation data, using only word embeddings, and achieved near equal performance with simple one-hot word vectors. However, the model performed considerably worse on the scope resolution task.

Serveega et al. (Sergeeva et al., 2019) recognized the dependency of neural network-based models on gold cue information, and proposed a BiLSTM-based model that achieved acceptable performance without using cue inputs. However, they do use Part-Of-Speech (POS) tags and dependency tree features. They compared model performance with gold cues, predicted cues and no cues and concluded that gold cues lead to the best performance, with little difference between predicted cues and no cues. For the cue predictions, they used an hierarchical LSTM model. Another method that did not use cue inputs was proposed

Table 6: Performance of existing methods on two corpora.

Conan Doyle corpus (Morante and Daelemans, 2012)				
Approach	Method	Cue det. F1	Scope res. F1	Cue input
RB	Lexicon (de Albornoz et al., 2012)	90.26	76.03	Pred
	Lexicon (Ballesteros et al., 2012)	71.88	62.65	Pred
ML	Lexicon+SVM (Gyawali and Solorio, 2012)	85.77	76.23	Pred
	SVM (Read et al., 2012)	92.10	85.26	Pred
	MRS Crawler (Packard et al., 2014)	-	86.6	Gold
			82.4	Pred*
CRF	CRF (Abu-Jbara and Radev, 2012)	90.98	82.70	Pred
	CRF (White, 2012)	90.00	83.51	Pred
NN	BiLSTM (Fancellu et al., 2016)	-	88.72	Gold
	NegBERT (Khandelwal and Sawant, 2020)	**92.94**	**92.36**	Gold

BioScope Abstracts corpus (Vincze et al., 2008)				
Approach	Method	Cue det. F1	Scope res. F1	Cue input
ML	Memory-based (Morante et al., 2008)	91.54	88.40	Gold
			80.99	Pred
	Metalearner (Morante and Daelemans, 2009)	**99.37**	90.67	Gold
			82.60	Pred
NN	CNN (Qian et al., 2016)	-	89.91	Gold
	BiLSTM+CRF (Fancellu et al., 2017)	-	92.11	Gold
	BiLSTM (Taylor and Harabagiu, 2018)	NR	88.85	None
	NegBERT (Khandelwal and Sawant, 2020)	95.65	**95.68**	Gold

Note: RB = Rule-based, ML = Machine Learning, CRF = Conditional Random Field, NN = Neural Networks. NR = Not Reported, a dash indicates that no cue detection was performed. *Predictions from SVM (Read et al., 2012).

by Taylor and Harabagiu (Taylor and Harabagiu, 2018). They tackled both tasks simultaneously with a cue/outside/inside labeling scheme and showed that the BiLSTM still correctly identified 89.02% of the scope tokens.

A.2 Motivation of the scope labeling scheme

The scope labeling scheme was motivated by the transition scores in a CRF model. Let $T^{5\times5}$ be a matrix such that $T_{i,j}$ represents a score associated with predicting label i for t_k and label j for t_{k+1}. Based on the structure of a scope within a sentence, we could expect the following kind of structure within T, where -2 = impossible, -1 = unlikely, 1 = likely, 2 = very likely:

$$T = \begin{array}{cc} & \begin{array}{cccc} \mathbf{O} & \mathbf{B} & \mathbf{C} & \mathbf{A} \end{array} \\ \begin{array}{c} \mathbf{O} \\ \mathbf{B} \\ \mathbf{C} \\ \mathbf{A} \end{array} & \begin{pmatrix} 1 & 1 & 1 & -2 \\ -2 & 1 & 1 & -2 \\ -1 & -2 & -1 & 2 \\ 1 & -2 & -2 & 1 \end{pmatrix} \end{array}$$