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

Daan de Jong and Ayoub Bagheri(✉)

Department of Methodology and Statistics, Utrecht University, Utrecht, The Netherlands
{d.dejong,a.bagheri}@uu.nl

Abstract. Negation is a complex grammatical phenomenon that has received considerable attention in the biomedical natural language processing domain. While neural network-based methods are the state-of-the-art in negation scope resolution, they often use the unrealistic assumption that negation 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. To tackle this issue, in this study, we adopted a two-step negation resolving approach to assess whether a neural network-based model, here a bidirectional long short-term memory, can be a an alternative for cue detection. Furthermore, we investigate how inaccurate cue predictions would affect the scope resolution performance. We ran various experiments on the open access Bio-Scope corpus. Experimental 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 subcorpus. We also found 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 and extensions with a post-processing algorithm. We advocate for more research into the application of automated deep learning on the effect of imperfect information on scope resolution.

Keywords: Negation cue detection · Negation scope resolution · Bi-directional long short-term memory · LSTM · Conditional random field

1 Introduction

Negations play an important role in the semantic representation of biomedical text, because they reverse the truth value of propositions [1]. Therefore, correct negation handling is a crucial step whenever the goal is to derive factual knowledge from biomedical text. There are two distinguish 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 [2–7]. Negation resolving shifts the focus towards the token level by approaching the problem as a sequence labeling task [8]. 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

P. Rosso et al. (Eds.): NLDB 2022, LNCS 13286, pp. 413–424, 2022.
https://doi.org/10.1007/978-3-031-08473-7_38

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 [neither
apples nor bananas are blue]."

Several studies adopted neural network-based approaches to resolve negations [10, 12,16]. 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. [9] 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 does a neural network-based model perform on the cue detection task?
2. How does a neural network-based model perform on the scope resolution task with imperfect cue information?

This study addresses these questions by applying a Bi-directional Long Short-Term Memory (BiLSTM) model [10] 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 subcorpora [11].

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 [8] with a Conditional Random Field (CRF) layer [12], in our experiments.

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 [14], we use the following labeling scheme for the cue detection task: For $k = 1, \ldots, n$, token t_k 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
- **NC** if it is not annotated as a cue

The scope label of token t_k is

- **O** if it is outside of the negation cue scope
- **B** if it is inside the negation scope, *before* the first cue token

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

– **C** if it is the first cue token in the scope
– **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 $\mathbf{c}$. Table 1 shows an example for the token sequence $\mathbf{t}$ with gold cue and scope labels for a given sentence: "It had [**no** effect on IL-10 secretion]."

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 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 (Fig. 1). For Task 1, we define a baseline model with an embedding layer and a softmax. For both tasks, we add a model where the softmax layer is replaced by a CRF layer to obtain a joint prediction for the token sequence.

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

[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 $k_L < k < k_R$, and discontinuous otherwise.

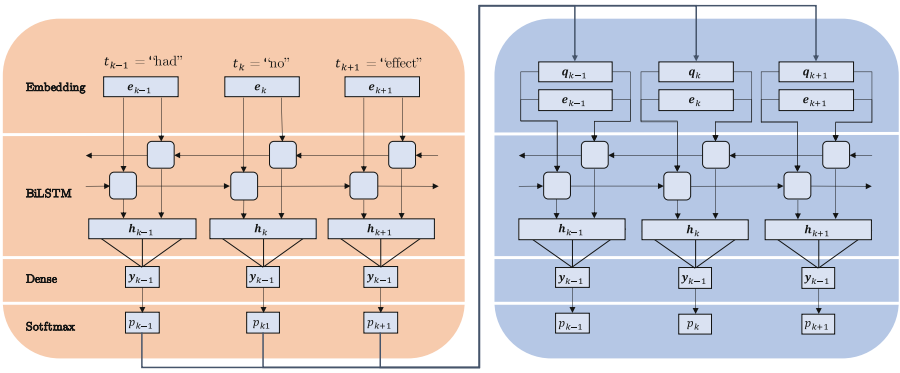

Fig. 1. Schematic representation of the BiLSTM model for cue detection (left) and scope resolution (right), for the example sentence "It had no effect on IL-10 secretion." at $k = 3$.

BioWordVec [18] 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 [19] on the MIMIC-III corpus [20]. 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 [21] 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 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 [22] 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 multiword 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:

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

[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 $x \mapsto (e^x - e^{-x})/(e^x + e^{-x})$.

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 **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 $p_k = \mathrm{softmax}(\mathbf{y}_k)$.

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

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

for the BiLSTM-CRF models, this likelihood is

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

Hyperparameters. The models were compiled and fitted with the Keras functional API for TensorFlow 2.3.1 in Python 3.7.6. 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 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.

Table 2. Descriptive statistics of the subcorpora.

	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.

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 [8]. 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 subcorpora from the open access BioScope corpus [11]. Together, these subcorpora 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, each such copy corresponded to a different negation instance. This resulted in 2,094 (14.48%) negation instances. A description of the subcorpora is provided in Table 2.

Tokenization. Biomedical text data poses additional challenges to the problem of tokenization [24]. 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 [25], 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 subcorpora. Tokenized sentences were truncated (23 sentences) or post-padded to match a length of 100 tokens.

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 is 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

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 [15]	**100**	**98.75**	**99.37**	**98.68**
NegBERT [14]	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 [15]	**100**	**95.72**	**96.08**	**92.15**
NegBERT [14]	NR	NR	90.23	NR

Note: PECM = Percentage Exact Cue Matches.

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 transferable to genres where the annotation style is different. For example, discontinuous scopes are quite common in the Conan Doyle corpus [13].

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. [12]. 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 $k + 2$.

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

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 [15]	90.68	90.68	90.67	73.36	100
	RecurCRFs* [17]	**94.9**	90.1	93.6	**92.3**	–
	NegBERT [14]	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 [15]	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 [15]	84.47	**84.95**	84.71	**50.26**	100
	NegBERT [14]	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 [15]	**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.

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.

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 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

1. Agirre, E., Bos, J., Diab, M., Manandhar, S., Marton, Y., Yuret, D.: * 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) (2012)
2. Mutalik, P.G., Deshpande, A., Nadkarni, P.M.: Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. J. Am. Med. Inform. Assoc. **8**(6), 598–609 (2001)
3. Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F., Buchanan, B.G.: A simple algorithm for identifying negated findings and diseases in discharge summaries. J. Biomed. Inform. **34**(5), 301–310 (2001)
4. Huang, Y., Lowe, H.J.: A novel hybrid approach to automated negation detection in clinical radiology reports. J. Am. Med. Inform. Assoc. **14**(3), 304–311 (2007)
5. Peng, Y., Wang, X., Lu, L., Bagheri, M., Summers, R., Lu, Z.: NegBio: a high-performance tool for negation and uncertainty detection in radiology reports. In: AMIA Summits on Translational Science Proceedings, p. 188 (2018)
6. Chen, L.: Attention-based deep learning system for negation and assertion detection in clinical notes. Int. J. Artif. Intell. Appl. (IJAIA) **10**(1) (2019)
7. Sykes, D., et al.: Comparison of rule-based and neural network models for negation detection in radiology reports. Nat. Lang. Eng. **27**(2), 203–224 (2021)
8. Morante, R., Liekens, A., Daelemans, W.: Learning the scope of negation in biomedical texts. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 715–724, October 2008
9. Read, J., Velldal, E., Øvrelid, L., Oepen, S.: 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), pp. 310–318 (2012)

10. Fancellu, F., Lopez, A., Webber, B.: Neural networks for negation scope detection. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 495–504, August 2016
11. Vincze, V., Szarvas, G., Farkas, R., Móra, G., Csirik, J.: The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. BMC Bioinform. **9**(11), 1–9 (2008)
12. Fancellu, F., Lopez, A., Webber, B., He, H.: 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, pp. 58–63, April 2017
13. Morante, R., Daelemans, W.: ConanDoyle-neg: annotation of negation in Conan Doyle stories. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, Istanbul, pp. 1563–1568, May 2012
14. Khandelwal, A., Sawant, S.: NegBERT: a transfer learning approach for negation detection and scope resolution. arXiv preprint arXiv:1911.04211 (2019)
15. Morante, R., Daelemans, W.: A metalearning approach to processing the scope of negation. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009), pp. 21–29, June 2009
16. Lazib, L., Qin, B., Zhao, Y., Zhang, W., Liu, T.: A syntactic path-based hybrid neural network for negation scope detection. Front. Comp. Sci. **14**(1), 84–94 (2018). https://doi.org/10.1007/s11704-018-7368-6
17. Fei, H., Ren, Y., Ji, D.: Negation and speculation scope detection using recursive neural conditional random fields. Neurocomputing **374**, 22–29 (2020)
18. Chen, Q., Peng, Y., Lu, Z.: BioSentVec: creating sentence embeddings for biomedical texts. In: 2019 IEEE International Conference on Healthcare Informatics (ICHI), pp. 1–5. IEEE, June 2019
19. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
20. Johnson, A.E., et al.: MIMIC-III, a freely accessible critical care database. Sci. Data **3**(1), 1–9 (2016)
21. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
22. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data (2001)
23. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
24. Díaz, N.P.C., Lóspez, M.J.M.: An analysis of biomedical tokenization: problems and strategies. In: Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis, pp. 40–49, September 2015
25. Loper, E., Bird, S.: NLTK: the natural language toolkit. arXiv preprint cs/0205028 (2002)
26. Peters, M.E., et al.: Deep contextualized word representations. arXiv 2018. arXiv preprint arXiv:1802.05365, December 2018
27. Banjade, R., Rus, V.: DT-Neg: tutorial dialogues annotated for negation scope and focus in context. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pp. 3768–3771, May 2016
28. Wang, T., Chen, P., Amaral, K., Qiang, J.: An experimental study of LSTM encoder-decoder model for text simplification. arXiv preprint arXiv:1609.03663 (2016)