Neural Networks For Negation Scope Detection

Neural Networks For Negation Scope Detection

Federico Fancellu and **Adam Lopez** and **Bonnie Webber**
School of Informatics
University of Edinburgh
11 Crichton Street, Edinburgh
f.fancellu[at]sms.ed.ac.uk, {alopez,bonnie}[at]inf.ed.ac.uk

Abstract

Automatic negation scope detection is a task that has been tackled using different classifiers and heuristics. Most systems are however 1) highly-engineered, 2) English-specific, and 3) only tested on the same genre they were trained on. We start by addressing 1) and 2) using a neural network architecture. Results obtained on data from the *SEM2012 shared task on negation scope detection show that even a simple feed-forward neural network using word-embedding features alone, performs on par with earlier classifiers, with a bi-directional LSTM outperforming all of them. We then address 3) by means of a specially-designed synthetic test set; in doing so, we explore the problem of detecting the negation scope more in depth and show that performance suffers from genre effects and differs with the type of negation considered.

1 Introduction

Amongst different extra-propositional aspects of meaning, negation is one that has received a lot of attention in the NLP community. Previous work have focused in particular on automatically detecting the *scope of negation*, that is, given a negative instance, to identify which tokens are affected by negation (§2). As shown in (1), only the first clause is negated and therefore we mark *he* and *the car*, along with the predicate *was driving* as inside the scope, while leaving the other tokens outside.

(1) He was **not** driving the car and she left to go home.

In the BioMedical domain there is a long line of research around the topic (e.g. Velldal et al. (2012) and Prabhakaran and Boguraev (2015)), given the importance of recognizing negation for information extraction from medical records. In more general domains, efforts have been more limited and most of the work centered around the *SEM2012 shared task on automatically detecting negation (§3), despite the recent interest (e.g. machine translation (Wetzel and Bond, 2012; Fancellu and Webber, 2014; Fancellu and Webber, 2015)).

The systems submitted for this shared task, although reaching good overall performance are highly feature-engineered, with some relying on heuristics based on English (Read et al. (2012)) or on tools that are available for a limited number of languages (e.g. Basile et al. (2012), Packard et al. (2014)), which do not make them easily portable across languages. Moreover, the performance of these systems was only assessed on data of the same genre (stories from Conan Doyle's *Sherlock Holmes*) but there was no attempt to test the approach on data of different genre.

Given these shortcomings, we investigate whether neural network based sequence-to-sequence models (§ 4) are a valid alternative. The first advantage of neural networks-based methods for NLP is that we could perform classification by means of unsupervised word-embeddings features only, under the assumption that they also encode structural information previous system had to explicitly represent as features. If this assumption holds, another advantage of continuous representations is that, by using a bilingual word-embedding space, we would be able to transfer the model cross-lingually, obviating the problem of the lack of annotated data in other languages.

The paper makes the following contributions:

1. *Comparable or better performance*: We show that neural networks perform on par with previously developed classifiers, with a bi-directional LSTM outperforming them

when tested on data from the same genre.

2. *Better understanding of the problem*: We analyze in more detail the difficulty of detecting negation scope by testing on data of different genre and find that the performance of word-embedding features is comparable to that of more fine-grained syntactic features.

3. *Creation of additional resources*: We create a *synthetic* test set of negative sentences extracted from Simple English Wikipedia (§ 5) and annotated according to the guidelines released during the *SEM2012 shared task (Morante et al., 2011), that we hope will guide future work in the field.

2 The task

Before formalizing the task, we begin by giving some definitions. A *negative sentence n* is defined as a vector of words $\langle w_1, w_2...w_n \rangle$ containing one or more negation **cues**, where the latter can be a word (e.g. *not*), a morpheme (e.g. *im*-patient) or a multi-word expression (e.g. *by no means, no longer*) inherently expressing negation.

A word is a **scope token** if included in the scope of a negation cue. Following Blanco and Moldovan (2011), in the *SEM2012 shared task the negation **scope** is understood as part of a knowledge representation focused around a negated event along with its related semantic roles and adjuncts (or its head in the case of a nominal event). This is exemplified in (2) (from Blanco and Moldovan (2011)) where the scope includes both the negated event *eat* along the subject *the cow*, the object *grass* and the PP *with a fork*.

(2) The cow did **n't** eat grass with a fork.[1]

Each cue defines its own *negation instance*, here defined as a tuple *I*(*n*,c) where $c \in \{1,0\}^{|n|}$ is a vector of length *n* s.t. $c_i = 1$ if w_i is part of the cue and 0 otherwise. Given *I* the goal of automatic scope detection is to predict a vector $s \in \{O,I\}^{|n|}$ s.t. s_i = I (inside of the scope) if w_i is in the scope of the cue or O (outside) otherwise.

In (3) for instance, there are two cues, *not* and *no longer*, each one defining a separate negation instance, *I1(n,c1)* and *I2(n,c2)*, and each with its own scope, *s1* and *s2*. In both (3a) and (3b), *n* = [I, do, not, love, you, and, you, are, no, longer, invited]; in (3a), the vector *c1* is 1 only at index 3 (w_2='not'), while in (3b) *c2* is 1 at position 9, 10 (where w_9 w_{10} = 'no longer'); finally the vectors *s1* and *s2* are I only at the indices of the words underlined and O anywhere else.

(3) a. I do **not** love you and you are no longer invited

b. I do not love you and you are **no longer** invited

There are the two main challenges involved in detecting the scope of negation: 1) a sentence can contain multiple instances of negation, sometimes nested and 2) scope can be discontinuous. As for 1), the classifier must correctly classify each word as being inside or outside the scope and assign each word to the correct scope; in (4) for instance, there are two negation cues and therefore two scopes, one spanning the entire sentence (3a.) and the other the subordinate only (3b.), with the latter being nested in the former (given that, according to the guidelines, if we negate the event in the main, we also negate its cause).

(4) a. I did **not** drive to school because my wife was not feeling well .[2]

b. I did not drive to school because my wife was **not** feeling well .

In (5), the classifier should instead be able to capture the long range dependency between the subject and its negated predicate, while excluding the positive VP in the middle.

(5) Naomi went to visit her parents to give them a special gift for their anniversary but **never** came back .

In the original task, the performance of the classifier is assessed in terms of precision, recall and F_1 measure over the number of words correctly classified as part of the scope (*scope tokens*) and over the number of scopes predicted that exactly

[1] In the *SEM2012 shared task, negation is not considered as a downward monotone function and definite expressions are included in its scope.

[2] One might object that the scope only spans over the subordinate given that it is the part of the scope most likely to be interpreted as false (*It is not the case that I drove to school because my wife was not at home, but for other reasons*). In the *SEM2012 shared task however this is defined separately as the *focus* of negation and considered as part of the scope. One reason to distinguish the two is the high ambiguity of the focus: one can imagine for instance that if the speaker stresses the words *to school* this will be most likely considered the focus and the statement interpreted as *It is not the case that I drive to school because my wife was not feeling well* (but I drove to the hospital instead).

match the gold scopes (*exact scope match*). As for latter, recall is a measure of accuracy since we score how many scopes we fully predict (true positives) over the total number of scopes in our test set (true positives and false negatives); precision takes instead into consideration false positives, that is those negation instances that are predicted as having a scope but in reality don't have any. This is the case of the interjection *No* (e.g. '**No**, leave her alone') that never take scope.

3 Previous work

Table 1 summarizes the performance of systems previously developed to resolve the scope of negation in non-Biomedical texts.

In general, supervised classifiers perform better than rule-based systems, although it is a combination of hand-crafted heuristics and SVM rankers to achieve the best performance. Regardless of the approach used, the syntactic structure (either constituent or dependency-based) of the sentence is often used to detect the scope of negation. This is because the position of the cue in the tree along with the projection of its parent/governor are strong indicators of scope boundaries. Moreover, given that during training we basically learn which syntactic patterns the scope are likely to span, it is also possible to hypothesize that this system should scale well to other genre/domain, as long as we can have a parse for the sentence; this however was never confirmed empirically. Although informative, these systems suffers form three main shortcomings: 1) they are highly-engineered (as in the case of Read et al. (2012)) and syntactic features add up to other PoS, word and lemma n-gram features, 2) they rely on the parser producing a correct parse and 3) they are English specific.

Other systems (Basile et al., 2012; Packard et al., 2014) tried to traverse a semantic representation instead. Packard et al. (2014) achieves the best results so far, using hand-crafted heuristics to traverse the MRS (Minimal Recursion Semantics) structures of negative sentences. If the semantic parser cannot create a reliable representation for a sentence, the system 'backs-off' to the hybrid model of Read et al. (2012), which uses syntactic information instead. This system suffers however from the same shortcomings mentioned above, in particular, given that MRS representation can only be built for a small set of languages.

4 Scope detection using Neural Networks

In this paper, we experiment with two different neural networks architecture: a one hidden layer **feed-forward neural network** and a **bi-directional LSTM** (Long Short Term Memory, BiLSTM below) model. We chose to 'start simple' from a feed-forward network to investigate whether even a simple model can reach good performance using word-embedding features only. We then turned to a BiLSTM because a better fit for the task. BiLSTM are sequential models that operate both in forward and backwards fashion; the backward pass is especially important in the case of negation scope detection, given that a scope token can appear in a string before the cue and it is therefore important that we see the latter first to classify the former. We opted in this case for LSTM over RNN cells given that their inner composition is able to better retain useful information when backpropagating the error.[4]

Both networks take as input a single negative instance $I(n,c)$. We represent each word $w_i \in n$ as a d-dimensional word-embedding vector $\mathbf{x} \in \mathbb{R}^d$ (d=50). In order to encode information about the cue, each word is also represented by a *cue*-embedding vector $\mathbf{c} \in \mathbb{R}^d$ of the same dimensionality of $\mathbf{x}$. $\mathbf{c}$ can only take two representations, *cue*, if c_i=1, or *notcue* otherwise. We also define $\mathbf{E}_w^{v \times d}$ as the word-embedding matrix, where v is the vocabulary size, and $\mathbf{E}_c^{2 \times d}$ as the cue-embedding matrix.

In the case of a feed-forward neural network, the input for each word $w_i \in n$ is the concatenation of its representation with the ones of its neighboring words in a context window of length l. This is because feed-forward networks treat the input units as separate and information about how words are arranged as sequences must be explicitly encoded in the input. We define these concatenations $\mathbf{x}_{conc}$ and $\mathbf{c}_{conc}$ as $\mathbf{x}_{w_{i-l}}...\mathbf{x}_{w_{i-1}}$; $\mathbf{x}_{w_i}$; $\mathbf{x}_{w_{i+1}}...\mathbf{x}_{w_i+l}$ and $\mathbf{c}_{w_{i-l}}...\mathbf{c}_{w_{i-1}}$; $\mathbf{c}_{w_i}$; $\mathbf{c}_{w_{i+1}}...\mathbf{c}_{w_{i+l}}$ respectively. We chose the value of l after analyzing the negation scopes in the dev set. We found that although the furthest scope tokens are 23 and 31 positions away from the cue on the left and the right respectively, 95% of the scope tokens fall in a window of 9 tokens to the left and 15 to the right, these two values being the window sizes we con-

[4] For more details on LSTM and related mathematical formulations, we refer to reader to Hochreiter and Schmidhuber (1997)

			Method	Scope tokens[3]			Exact scope match		
				Prec.	*Rec.*	F_1	*Prec.*	*Rec.*	F_1
*SEM2012	Closed track	UiO1 (Read et al., 2012)	heuristics + SVM	81.99	88.81	85.26	87.43	61.45	72.17
		UiO2 (Lapponi et al., 2012)	CRF	86.03	81.55	83.73	85.71	62.65	72.39
		FBK (Chowdhury and Mahbub, 2012)	CRF	81.53	82.44	81.89	88.96	58.23	70.39
		UWashington (White, 2012)	CRF	83.26	83.77	83.51	82.72	63.45	71.81
		UMichigan (Abu-Jbara and Radev, 2012)	CRF	84.85	80.66	82.70	90.00	50.60	64.78
		UABCoRAL (Gyawali and Solorio, 2012)	SVM	85.37	68.86	76.23	79.04	53.01	63.46
	Open track	UiO2 (Lapponi et al., 2012)	CRF	82.25	82.16	82.20	85.71	62.65	72.39
		UGroningen (Basile et al., 2012)	rule-based	69.20	82.27	75.15	76.12	40.96	53.26
		UCM-1 (de Albornoz et al., 2012)	rule-based	85.37	68.53	76.03	82.86	46.59	59.64
		UCM-2 (Ballesteros et al., 2012)	rule-based	58.30	67.70	62.65	67.13	38.55	48.98
Packard et al. (2014)			heuristics + SVM	**86.1**	**90.4**	**88.2**	**98.8**	**65.5**	**78.7**

Table 1: Summary of previous work on automatic detection of negation scope.

sider for our input. The probability of a given input is then computed as follows:

$$\begin{aligned} \mathbf{h} &= \sigma(\mathbf{W}_x\mathbf{x}_{conc} + \mathbf{W}_c\mathbf{c}_{conc} + \mathbf{b}) \\ y &= g(\mathbf{W}_y\mathbf{h} + \mathbf{b}_y) \end{aligned} \tag{1}$$

where $\mathbf{W}$ and $\mathbf{b}$ the weight and biases matrices, $\mathbf{h}$ the hidden layer representation, σ the sigmoid activation function and g the softmax operation ($g(z_m) = e^{z_m} / \sum_k e^{z_k}$) to assign a probability to the input of belonging to either the inside (I) or outside (O) of the scope classes.

In the biLSTM, no concatenation is performed, given that the structure of the network is already sequential. The input to the network for each word w_i are the word-embedding vector $\mathbf{x}_{w_i}$ and the cue-embedding vector $\mathbf{c}_{w_i}$, where w_i constitutes a time step. The computation of the hidden layer at time t and the output can be represented as follows:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_x^{(i)}\mathbf{x} + \mathbf{W}_c^{(i)}\mathbf{c} + \mathbf{W}_h^{(i)}\mathbf{h_{t-1}} + \mathbf{b}^{(i)}) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_x^{(f)}\mathbf{x} + \mathbf{W}_c^{(f)}\mathbf{c} + \mathbf{W}_h^{(f)}\mathbf{h_{t-1}} + \mathbf{b}^{(f)}) \\ \mathbf{o}_t &= \sigma(\mathbf{W}_x^{(o)}\mathbf{x} + \mathbf{W}_c^{(o)}\mathbf{c} + \mathbf{W}_h^{(o)}\mathbf{h_{t-1}} + \mathbf{b}^{(o)}) \\ \tilde{\mathbf{c}}_t &= tanh(\mathbf{W}_x^{(c)}\mathbf{x} + \mathbf{W}_c^{(c)}\mathbf{c} + \mathbf{W}_h^{(c)}\mathbf{h_{t-1}} + \mathbf{b}^{(c)}) \\ \mathbf{c}_t &= \mathbf{f}_t \cdot \tilde{\mathbf{c}}_{t-1} + \mathbf{i}_t \cdot \tilde{\mathbf{c}}_t \\ \mathbf{h}_{back/forw} &= \mathbf{o}_t \cdot tanh(\mathbf{c}_t) \\ y_t &= g(\mathbf{W}_y(\mathbf{h}_{back}; \mathbf{h}_{forw}) + \mathbf{b}_y) \end{aligned} \tag{2}$$

where the $\mathbf{W}$s are the weight matrices, $\mathbf{h}_{t-1}$ the hidden layer state a time t-1, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ the input, forget and the output gate at the time t and $\mathbf{h}_{back}$; $\mathbf{h}_{forw}$ the concatenation of the backward and forward hidden layers.

Finally, in both networks our training objective is to minimise, for each negative instance, the negative log likelihood *J(W,b)* of the correct predictions over gold labels:

$$\begin{aligned} J(W,b) = -\frac{1}{l}\sum_{i=1}^{l} & y^{(w_i)} \log h_\theta(x^{(w_i)}) \\ & + (1 - y^{(w_i)}) \log(1 - h_\theta(x^{(w_i)})) \end{aligned} \tag{3}$$

where l is the length of the sentence $n \in I$, $x^{(w_i)}$ the probability for the word w_i to belong to either the I or O class and $y^{(w_i)}$ its gold label.

An overview of both architectures is shown in Figure 1.

4.1 Experiments

Training, development and test set are a collection of stories from Conan Doyle's *Sherlock Holmes* annotated for cue and scope of negation and released in concomitance with the *SEM2012 shared task.[5] For each word, the correspondent lemma, POS tag and the constituent subtree it belongs to are also annotated. If a sentence contains multiple instances of negation, each is annotated separately.

Both training and testing is done on negative sentences only, i.e. those sentences with at least one cue annotated. Training and test size are of 848 and 235 sentences respectively. If a sentence contains multiple negation instances, we create as many copies as the number of instances. If the sentence contains a morphological cue (e.g. *im*patient) we split it into affix (**im-**) and root (*patient*), and consider the former as cue and the latter as part of the scope.

Both neural network architectures are implemented using TensorFlow (Abadi et al., 2015) with a 200-units hidden layer (400 in total for two concatenated hidden layers in the BiLSTM), the Adam optimizer (Kingma and Ba, 2014) with a

[5]For the statistics regarding the data, we refer the reader to Morante and Blanco (2012).

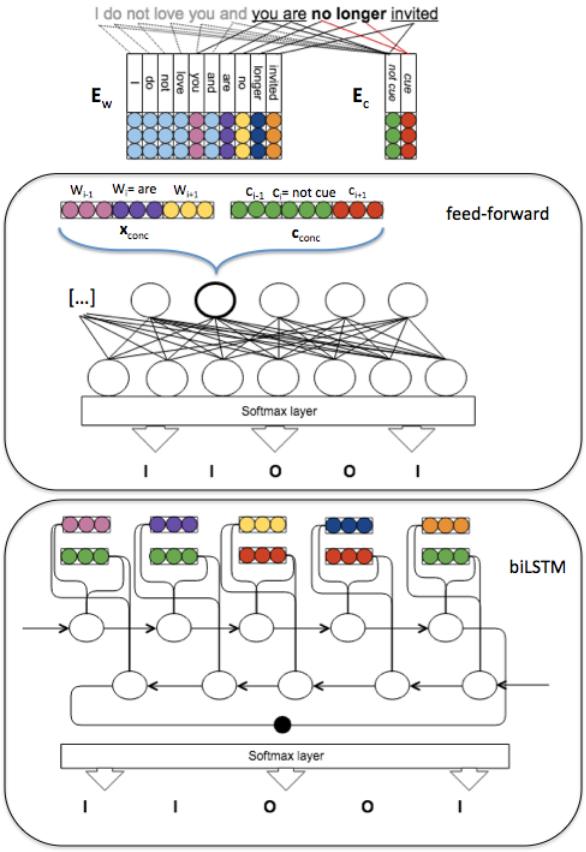

Figure 1: An example of scope detection using feed-forward and BiLSTM for the tokens '*you are* **no longer** *invited*' in the instance in ex. (3b).

starting learning rate of 0.0001, learning rate decay after 10 iterations without improvement and early stopping. In both cases we experimented with different settings:

1. **Simple baseline**: In order to understand how hard the task of negation scope detection is, we created a simple baseline by tagging as part of the scope all the tokens 4 words to the left and 6 to the right of the cue; these values were found to be the average span of the scope in either direction in the training data.

2. **Cue info (C)**: The word-embedding matrix is randomly initialised and updated relying on the training data only. Information about the cue is fed through another set of embedding vectors, as shown in 4. This resembles the 'Closed track' of the *SEM2012 shared task since no external resource is used.

3. **Cue info + external embeddings (E)**: This is the same as setting (2) except that the embeddings are pre-trained using external data. We experimented with both keeping the word-embedding matrix fixed and updating it during training but we found small or no difference between the two settings. To do this, we train a word-embedding matrix using Word2Vec (Mikolov et al., 2013) on 770 million tokens (for a total of 30 million sentences and 791028 types) from the 'One Billion Words Language Modelling' dataset [6] and the Sherlock Holmes data (5520 sentences) combined. The dataset was tokenized and morphological cues split into negation affix and root to match the Conan Doyle's data. In order to perform this split, we matched each word against an hand-crafted list of words containing affixal negation[7]; this method have an accuracy of 0.93 on the Conan Doyle test data.

4. **Adding PoS / Universal PoS information (PoS/uni PoS)**: This was mainly to assess whether we could get further improvement by adding additional information. For all the setting above, we also add an extra embedding input vector for the POS or Universal POS of each word w_i. As for the word and the cue embeddings, PoS-embedding information are fed to the hidden layer through a separate weight matrix. When pre-trained, the training data for the external PoS-embedding matrix is the same used for building the word embedding representation, except that in this case we feed the PoS / Universal PoS tag for each word. As in (3), we experimented with both updating the tag-embedding matrix and keeping it fixed but found again small or no difference between the two settings. In order to maintain consistency with the original data, we perform PoS tagging using the GENIA tagger (Tsuruoka et al., 2005)[8] and then map the resulting tags to universal POS tags.[9]

4.2 Results

The results for the scope detection task are shown in Table 2.

[6] Available at `https://code.google.com/archive/p/word2vec/`

[7] The list was courtesy of Ulf Hermjakob and Nathan Schneider.

[8] `https://github.com/saffsd/geniatagger`

[9] Mapping available at `https://github.com/slavpetrov/universal-pos-tags`

Results for both architecture when word-embedding features *only* are used (C and C + E) show that neural networks are a valid alternative for scope detection, with bi-directional LSTM being able to outperform all previously developed classifiers on both scope token recognition and exact scope matching. Moreover, a bi-directional LSTM shows similar performance to the hybrid system of Packard et al. (2014) (rule-based + SVM as a back-off) in absence of any hand-crafted heuristics.

It is also worth noticing that although pre-training the word-embedding and PoS-embedding matrices on external data leads to a slight improvement in performance, the performance of the systems using internal data only is already competitive; this is a particularly positive result considering that the training data is relatively small.

Finally, adding universal POS related information leads to a better performance in most cases. The fact that the best system is built using language-independent features only is an important result when considering the portability of the model across different languages.

4.3 Error analysis

In order to understand the kind of errors our best classifier makes, we performed an error analysis on the held-out set.

First, we investigate whether the per-instance prediction accuracy correlates with scope-related (length of the scope to the left, to the right and combined; maximum length of the gap in a discontinuous scope) and cue-related (type of cue -one-word, prefixal, suffixal, multiword-) variables. We also checked whether the neural network is biased towards the words it has seen in the training(for instance, if it has seen the same token always labeled as O it will then classify it as O). For our best biLSTM system, we found only weak to moderate negative correlations with the following variables:

- *length of the gap*, if the scope is discontinuous (r=-0.1783, p = 0.004);
- *overall scope length* (r=-0.3529, $p < 0.001$);
- *scope length to the left and to the right* (r=-0.3251 and -0.2659 respectively with $p < 0.001$)
- *presence of a prefixal cue* (r=-0.1781, p = 0.004)
- *presence of a multiword cue* (r=-0.1868, p = 0.0023)

meaning that the variables considered are not strong enough to be considered as error patterns.

For this reason we also manually analyzed the 96 negation scopes that the best biLSTM system predicted incorrectly and noticed several error patterns:

- in 5 cases, the scope should only span on the subordinate but end up including elements from the main. In (6) for instance, where the system prediction is reported in curly brackets, the BiLSTM ends up including the main predicate with its subject in the scope.

  (6) You felt so strongly about it that {I knew you could} **not** {think of Beecher without thinking of that also} .

- in 5 cases, the system makes an incorrect prediction in presence of the syntactic inversion, where a subordinate appears before the main clause; in (7) for instance, the system extends the prediction to the main clause when the scope should instead span the subordinate only.

  (7) But {if she does} **not** {wish to shield him she would give his name}

- in 8 cases, where two VPs, one positive and one negative, are coordinated, the system ends up including in the scope the positive VP as well, as shown in (8). We hypothesized this is due to the lack of such examples in the training set.

  (8) Ah, {you do} **n't** {know Sarah 's temper or you would wonder no more} .

As in Packard et al. (2014), we also noticed that in 15 cases, the gold annotations do not follow the guidelines; in the case of a negated adverb in particular, as shown in (9a) and (9b) the annotations do not seem to agree on whether consider as scope only the adverb or the entire clause around it.

	Scope tokens							Exact scope match		
System	gold	tp	fp	fn	*Prec.*	*Rec.*	F_1	*Prec.*	*Rec.*	F_1
Baseline	1830	472	3031	1358	13.47	25.79	17.70	0.0	0.0	0.0
Best closed track: UiO1	N/A	N/A	N/A	N/A	81.99	88.81	85.26	87.43	61.45	72.17
Packard et al. (2014)	N/A	N/A	N/A	N/A	86.1	**90.4**	88.2	98.8	**65.5**	**78.7**
FF - C	1830	1371	273	459	83.39	74.91	78.92	93.61	34.10	50.00
FF - C + PoS	1830	1413	235	417	85.74	77.21	81.25	92.51	37.50	53.33
FF - C + Uni PoS	1830	1435	276	395	83.86	78.41	81.05	93.06	36.57	52.51
FF - C + E	1830	1455	398	375	78.52	79.50	79.01	89.53	30.19	45.16
FF - C + PoS + E	1830	1413	179	417	88.75	77.21	82.58	96.63	44.23	60.68
FF - C + Uni PoS + E	1830	1412	158	418	89.93	77.15	83.05	96.58	43.46	59.94
BiLSTM - C	1830	1583	175	247	90.04	86.50	88.23	98.71	58.77	73.68
BiLSTM - C + PoS	1830	1591	203	239	88.68	86.93	87.80	98.70	58.01	73.07
BiLSTM - C + Uni Pos	1830	**1592**	193	238	89.18	86.95	88.07	98.96	57.63	72.77
BiLSTM - C + E	1830	1570	157	260	90.90	85.79	88.27	99.37	60.83	75.47
BiLSTM - C + PoS + E	1830	1546	148	284	91.26	84.48	87.74	98.75	60.30	74.88
BiLSTM - C + Uni PoS + E	1830	1552	124	272	**92.62**	85.13	**88.72**	**99.40**	63.87	77.77

Table 2: Results for the scope detection task on the held-out set. Results are plotted against the simple baseline, the best system so far (Packard et al., 2014) and the system with the highest F_1 for *scope tokens* classification amongst the ones submitted for the *SEM2012 shared task. We also report the number of gold scope tokens, true positive (tp), false positives(fp) and false negatives(fn).

(9) a. [...] tossing rest**less**ly from side to side [..]

b. [...] glaring help**less**ly at the frightful thing which was hunting him down.

5 Evaluation on synthetic data set

5.1 Methodology

One question left unanswered by previous work is whether the performance of scope detection classifiers is robust against data of a different genre and whether different types of negation lead to difference in performance. To answer this, we compare two of our systems with the only original submission to the *SEM2012 we found available (White, 2012)[10]. We decided to use both our best system, BiLSTM+C+UniPoS+E and a sub-optimal systems, BiLSTM+C+E to also assess the robustness of non-English specific features.

The synthetic test set here used is built on sentences extracted from Simple Wikipedia and manually annotated for cue and scope according to the annotation guidelines released in concomitance with the *SEM2012 shared task (Morante et al., 2011). We created 7 different subsets to test different types of negative sentences:

Simple: we randomly picked 50 positive sentences, containing only one predicate, no dates and no named entities, and we made them negative by adding a negation cue (*do* support or minor morphological changes were added when required). If more than a lexical negation cue fit in the context, we used them all by creating more than one negative counterpart, as shown in (10). The sentences were picked to contain different kind of predicates (verbal, existential, nominal, adjectival).

(10) a. Many people disagree on the topic

b. Many people do **not** disagree on the topic

c. Many people **never** disagree on the topic

Lexical: we randomly picked 10 sentences[11] for each **lexical** (i.e. one-word) cue in training data (these are *not*, *no*, *none*, *nobody*, *never*, *without*)

Prefixal: we randomly picked 10 sentences for each prefixal cue in the training data (*un-*, *im-*, *in-*, *dis-*, *ir-*)

Suffixal: we randomly picked 10 sentences for the suffixal cue *-less*.

Multi-word: we randomly picked 10 sentences for each multi-word cue (*neither...nor*,*no longer*,*by no means*).

Unseen: we include 10 sentences for each of the negative prefixes *a-* (e.g. *a*-cyclic), *ab-* (e.g. *ab*-normal) *non-* (e.g. *non*-Communist) that are not annotated as cue in the Conan Doyle corpus,

[10] In order for the results to be comparable, we feed White's system with the cues from the gold-standard instead of automatically detecting them.

[11] In some cases, we ended up with more than 10 examples for some cues given that some of the sentences we picked contained more than a negation instance.

		Scope tokens							Exact scope match		
	Data	gold	tp	fp	fn	*Prec.*	*Rec.*	F_1	*Prec.*	*Rec.*	F_1
White (2012)	simple	850	**830**	0	20	100.00	97.65	**98.81**	100.00	93.98	**96.90**
	lexical	814	652	101	162	86.59	80.10	83.22	100.00	58.41	**73.75**
	prefixal	316	232	103	83	68.98	73.40	**71.12**	100.00	32.76	**49.35**
	suffixal	100	**78**	7	22	91.76	78.00	**84.32**	100.00	69.23	**81.82**
	multi-word	269	190	12	49	89.62	70.63	79.00	100.00	9.00	16.67
	unseen	220	138	40	82	77.53	62.73	69.35	100.00	38.89	**56.00**
	avg.	2569	**2120**	263	418	**85.74**	**77.08**	**80.97**	**100.00**	**50.37**	**62.41**
BiLSTM - C+ E	simple	850	827	0	23	100.00	97.29	98.62	100.00	88.72	94.02
	lexical	814	618	120	133	85.01	83.66	84.33	100.00	40.35	57.50
	prefixal	316	**235**	156	81	60.10	74.36	66.47	100.00	10.34	18.75
	suffixal	100	53	5	47	91.52	53.46	67.50	100.00	15.28	26.66
	multi-word	269	192	22	79	93.65	71.37	81.01	100.00	36.36	**53.00**
	unseen	220	151	79	69	66.09	69.05	67.54	100.00	22.22	36.36
	avg.	2569	2076	382	432	82.72	74.86	77.57	100.00	35.54	47.76
BiLSTM - C+ UniPos + E	simple	850	816	0	34	100.00	96	97.95	100.00	82.70	90.05
	lexical	814	**668**	97	146	87.32	82.06	**84.61**	100.00	42.10	59.25
	prefixal	316	231	128	85	64.34	73.10	68.44	100.00	20.68	34.28
	suffixal	100	54	3	47	94.73	53.46	68.35	100.00	38.46	55.55
	multi-word	269	**202**	19	67	91.40	75.09	**82.44**	100.00	27.27	42.85
	unseen	220	**152**	56	71	73.07	68.16	**70.53**	100.00	25.00	40.00
	avg.	2569	**2123**	303	449	85.14	74.64	78.72	100.00	39.36	53.66

Table 3: Results for the scope detection task on the synthetic test set.

to test whether the system can generalise the classification to unseen cues.[12]

5.2 Results

Table 3. shows the results for the comparison on the synthetic test set. The first thing worth noticing is that by using word-embedding features only it is possible to reach comparable performance with a classifier using syntactic features, with universal PoS generally contributing to a better performance; this is particularly evident in the *multi-word* and *lexical* sub-sets. In general, genre effects hinder both systems; however, considering that the training data is less than 1000 sentences, results are relatively good.

Performance gets worse when dealing with morphological cues and in particular in the case of our classifier, with suffixal cues; at a closer inspection however, the cause of such poor performance is attributable to a discrepancy between the annotation guidelines and the training data, already noted in §4.4. The guidelines state in fact that '*If the negated affix is attached to an adverb that is a complement of a verb, the negation scopes over the entire clause*'(Morante et al., 2011, p. 21) and we annotated suffixal negation in this way. However, 3 out of 4 examples of suffixal negation in adverbs in the training data (e.g. 9a.) mark the scope on the adverbial root only and that's what our classifiers learn to do.

Finally, it can be noticed that our system does worse at exact scope matching than the CRF classifier. This is because White (2012)'s CRF model is build on constituency-based features that will then predict scope tokens based on constituent boundaries (which, as we said, are good indicator of scope boundaries), while neural networks, basing the prediction only on word-embedding information, might extend the prediction over these boundaries or leave 'gaps' within.

6 Conclusion and Future Work

In this work, we investigated and confirmed that neural networks sequence-to-sequence models are a valid alternative for the task of detecting the scope of negation. In doing so we offer a detailed analysis of its performance on data of different genre and containing different types of negation, also in comparison with previous classifiers, and found that non-English specific continuous representation can perform batter than or on par with more fine-grained structural features.

Future work can be directed towards answering two main questions:

Can we improve the performance of our classifier? To do this, we are going to explore whether adding language-independent structural informa-

[12]The data, along with the code, is freely available at `https://github.com/ffancellu/NegNN`

tion (e.g. universal dependency information) can help the performance on exact scope matching.

Can we transfer our model to other languages? Most importantly, we are going to test the model using word-embedding features extracted from a bilingual embedding space.

Acknowledgments

This project was also founded by the European Unions Horizon 2020 research and innovation programme under grant agreement No 644402 (HimL).

The authors would like to thank Naomi Saphra, Nathan Schneider and Claria Vania for the valuable suggestions and the three anonymous reviewers for their comments.

References

M Abadi, A Agarwal, P Barham, E Brevdo, Z Chen, C Citro, GS Corrado, A Davis, J Dean, M Devin, et al. 2015. Tensorflow: Large-scale machine learning on heterogeneous systems. *White paper, Google Research*.

Amjad Abu-Jbara and Dragomir Radev. 2012. Umichigan: A conditional random field model for resolving the scope 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 328–334. Association for Computational Linguistics.

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 *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 288–293. Association for Computational Linguistics.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Ugroningen: Negation detection with discourse representation structures. 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 301–309. Association for Computational Linguistics.

Eduardo Blanco and Dan I Moldovan. 2011. Some issues on detecting negation from text. In *FLAIRS Conference*, pages 228–233. Citeseer.

Md Chowdhury and Faisal Mahbub. 2012. Fbk: Exploiting phrasal and contextual clues for negation scope detection. 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 340–346. Association for Computational Linguistics.

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 *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 282–287. Association for Computational Linguistics.

Federico Fancellu and Bonnie L Webber. 2014. Applying the semantics of negation to smt through n-best list re-ranking. In *EACL*, pages 598–606.

Federico Fancellu and Bonnie Webber. 2015. Translating negation: A manual error analysis. *ExProM 2015*, page 1.

Binod Gyawali and Thamar Solorio. 2012. Uabcoral: a preliminary study for resolving the scope 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 275–281. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. Uio 2: sequence-labeling negation using dependency features. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 319–327. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Roser Morante and Eduardo Blanco. 2012. * sem 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference*

on Lexical and Computational Semantics-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.

Roser Morante, Sarah Schrauwen, and Walter Daelemans. 2011. Annotation of negation cues and their scope: Guidelines v1. *Computational linguistics and psycholinguistics technical report series, CTRS-003*.

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 *ACL (1)*, pages 69–78.

Vinodkumar Prabhakaran and Branimir Boguraev. 2015. Learning structures of negations from flat annotations. *Lexical and Computational Semantics (* SEM 2015)*, page 71.

Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2012. Uio 1: Constituent-based discriminative ranking for negation resolution. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 310–318. Association for Computational Linguistics.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. *Advances in informatics*, pages 382–392.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers, and the role of syntax. *Computational linguistics*, 38(2):369–410.

Dominikus Wetzel and Francis Bond. 2012. Enriching parallel corpora for statistical machine translation with semantic negation rephrasing. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 20–29. Association for Computational Linguistics.

James Paul White. 2012. Uwashington: Negation resolution using machine learning methods. 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 335–339. Association for Computational Linguistics.