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 

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 Prabhkaran 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 $(w_1, w_2, \ldots, w_n)$ 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 *not* 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 $s \in \{O,1\}^{[n]}$ s.t. $s_j = 1$ (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, $I_1(n,c_1)$ and $I_2(n,c_2)$, and each with its own scope, $s_1$ and $s_2$. In both (3a) and (3b), $n = [I, do, not, love, you, and, you, are, no, longer, invited]; in (3a), the vector $c_1$ is 1 only at index 3 ($w_2 = ’not’)$, while in (3b) $c_2$ is 1 at position 9, 10 (where $w_9, w_{10} = ’no longer’$); finally the vectors $s_1$ and $s_2$ 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

---

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

²One 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 \(E_{w}^{\text{voc}}\) as the word-embedding matrix, where \(v\) is the vocabulary size, and \(E_{c}^{\text{voc}}\) 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 \(x_{\text{conc}}\) and \(c_{\text{conc}}\) as \(x_{w_{i-1}}, \ldots, x_{w_{i-l}}; x_{w_{i}}; x_{w_{i+1}}, \ldots, x_{w_{i+l}}\) and \(c_{w_{i-1}}, \ldots, c_{w_{i-l}}; c_{w_{i}}; c_{w_{i+1}}, \ldots, c_{w_{i+l}}\), 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-
The probability of a given input is then computed as follows:

\[
\begin{align*}
    h &= \sigma(W_x x_{\text{conc}} + W_c c_{\text{conc}} + b) \\
    y &= g(W_y h + b_y)
\end{align*}
\]

where \(W\) and \(b\) the weight and biases matrices, \(h\) the hidden layer representation, \(\sigma\) 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 \(w_t\) is the word-embedding vector \(x_{w_t}\) and the cue-embedding vector \(c_{w_t}\), where \(w_t\) constitutes a time step. The computation of the hidden layer at time \(t\) and the output can be represented as follows:

\[
\begin{align*}
    i_t &= \sigma(W_x^{(i)} x + W_c^{(i)} c + W_h^{(i)} h_{t-1} + b^{(i)}) \\
    f_t &= \sigma(W_x^{(f)} x + W_c^{(f)} c + W_h^{(f)} h_{t-1} + b^{(f)}) \\
    o_t &= \sigma(W_x^{(o)} x + W_c^{(o)} c + W_h^{(o)} h_{t-1} + b^{(o)}) \\
    c_t &= tanh(W_x^{(c)} x + W_c^{(c)} c + W_h^{(c)} h_{t-1} + b^{(c)}) \\
    h_{\text{back/forw}} &= o_t \cdot tanh(c_t) \\
    y_t &= g(W_y (h_{\text{back}}; h_{\text{forw}}) + b_y)
\end{align*}
\]

where the \(W_s\) are the weight matrices, \(h_{t-1}\) the hidden layer state a time \(t-1\), \(i_t, f_t, o_t\) the input, forget and the output gate at the time \(t\) and \(h_{\text{back/forw}}\) 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:

\[
J(W; b) = \frac{1}{l} \sum_{t=1}^{l} y^{(w_i)} \log h_\theta(x^{(w_i)}) + (1 - y^{(w_i)}) \log (1 - h_\theta(x^{(w_i)}))
\]

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*. For each word, the correspondent lemma, POS tag and the constituent subtree belong 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

---

1. 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 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; 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) and then map the resulting tags to universal POS tags.

4.2 Results

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

---

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

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

8https://github.com/saffsd/Geniatagger

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 **not** {if she does} **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, **not** {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.
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).

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). 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).

- **Lexical**: we randomly picked 10 sentences 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.

\[\text{(9) a. [...] tossing restlessy from side to side [...]}
\]
\[\text{b. [...] glaring helplessly at the frightful thing which was hunting him down.}
\]

\[\text{(10) a. Many people disagree on the topic}
\]
\[\text{b. Many people do not disagree on the topic}
\]
\[\text{c. Many people never disagree on the topic}
\]

**Lexical**: we randomly picked 10 sentences 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 each 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.
<table>
<thead>
<tr>
<th>Scope tokens</th>
<th>Data</th>
<th>gold</th>
<th>tp</th>
<th>fp</th>
<th>fn</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F</th>
<th>Exact scope match</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>White (2012)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>simple</td>
<td>850</td>
<td>380</td>
<td>0</td>
<td>20</td>
<td>100.00</td>
<td>97.65</td>
<td>98.81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lexical</td>
<td>814</td>
<td>652</td>
<td>101</td>
<td>162</td>
<td>86.59</td>
<td>80.10</td>
<td>83.22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prefixal</td>
<td>316</td>
<td>232</td>
<td>103</td>
<td>83</td>
<td>68.98</td>
<td>73.40</td>
<td>71.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>suffixal</td>
<td>100</td>
<td>78</td>
<td>7</td>
<td>22</td>
<td>91.76</td>
<td>78.00</td>
<td>84.32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>multi-word</td>
<td>269</td>
<td>190</td>
<td>12</td>
<td>49</td>
<td>89.62</td>
<td>70.63</td>
<td>79.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unseen</td>
<td>220</td>
<td>138</td>
<td>40</td>
<td>82</td>
<td>77.53</td>
<td>62.73</td>
<td>69.35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>avg.</td>
<td>2569</td>
<td>2120</td>
<td>263</td>
<td>418</td>
<td>85.74</td>
<td>77.08</td>
<td>80.97</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BiLSTM - C+ E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>simple</td>
<td>850</td>
<td>827</td>
<td>0</td>
<td>23</td>
<td>100.00</td>
<td>97.29</td>
<td>98.62</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lexical</td>
<td>814</td>
<td>618</td>
<td>120</td>
<td>133</td>
<td>85.01</td>
<td>83.66</td>
<td>84.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prefixal</td>
<td>316</td>
<td>235</td>
<td>5</td>
<td>47</td>
<td>91.52</td>
<td>53.46</td>
<td>66.47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>suffixal</td>
<td>100</td>
<td>72</td>
<td>5</td>
<td>47</td>
<td>94.73</td>
<td>53.46</td>
<td>68.35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>multi-word</td>
<td>269</td>
<td>192</td>
<td>22</td>
<td>79</td>
<td>93.65</td>
<td>71.37</td>
<td>81.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unseen</td>
<td>220</td>
<td>151</td>
<td>79</td>
<td>69</td>
<td>66.09</td>
<td>69.05</td>
<td>67.54</td>
<td></td>
</tr>
<tr>
<td></td>
<td>avg.</td>
<td>2569</td>
<td>2076</td>
<td>382</td>
<td>432</td>
<td>82.72</td>
<td>74.86</td>
<td>77.57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BiLSTM - C+ UniPos + E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>simple</td>
<td>850</td>
<td>816</td>
<td>0</td>
<td>34</td>
<td>100.00</td>
<td>96.97</td>
<td>97.95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>lexical</td>
<td>814</td>
<td>668</td>
<td>97</td>
<td>146</td>
<td>87.32</td>
<td>73.10</td>
<td>82.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prefixal</td>
<td>316</td>
<td>231</td>
<td>8</td>
<td>54</td>
<td>64.34</td>
<td>73.10</td>
<td>66.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>suffixal</td>
<td>100</td>
<td>54</td>
<td>3</td>
<td>47</td>
<td>94.73</td>
<td>53.46</td>
<td>68.35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>multi-word</td>
<td>269</td>
<td>202</td>
<td>19</td>
<td>67</td>
<td>91.40</td>
<td>70.59</td>
<td>82.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>unseen</td>
<td>220</td>
<td>152</td>
<td>76</td>
<td>67</td>
<td>73.07</td>
<td>68.16</td>
<td>70.53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>avg.</td>
<td>2569</td>
<td>2123</td>
<td>303</td>
<td>449</td>
<td>85.14</td>
<td>74.64</td>
<td>78.72</td>
<td></td>
</tr>
</tbody>
</table>

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

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 better 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-

---

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


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.


