Detecting negation scope is easy, except when it isn’t

Federico Fancellu\textsuperscript{1}  Adam Lopez\textsuperscript{1}  Bonnie Webber\textsuperscript{1}  Hangfeng He\textsuperscript{2}

\textsuperscript{1}ILCC, School of Informatics, University of Edinburgh
\textsuperscript{2}School of Electronics Engineering and Computer Science, Peking University

\texttt{f.fancellu@sms.ed.ac.uk, \{alopez, bonnie\}@inf.ed.ac.uk, hangfenghe@pku.edu.cn}
Negation Scope Detection (at the string level)

- **Input**: a sentence containing at least one negation marker (or cue)
- **Task**: classify a token as part of the **scope** of the cue or not (binary classification)

I am Italian but I do n’t eat pizza
Negation Scope Detection (at the string level)

- **Input**: a sentence containing at least one negation marker (or cue)
- **Task**: classify a token as part of the scope of the cue or not (binary classification)

I am Italian but I do n’t eat pizza

*It is not the case that I eat pizza*
Negation Scope Detection (at the string level)

- **Input**: a sentence containing at least one negation marker (or cue)
- **Task**: classify a token as part of the scope of the cue or not (binary classification)

I am Italian but I don’t eat pizza

*It is not the case that I eat pizza*

*It is the case that I am Italian*
Neural Networks for Negation Scope Detection [Fancellu et al., 2016]

- Bi-LSTM for negation scope detection
- Performance on par or better than previous heavily-engineered or heuristics-based approaches
- Tested on Conan-Doyle neg. [Morante et Daelemans, 2012]
This work

- Several corpora annotated with negation scope
  - Different annotation decisions
  - Different domains
- Our question: **Does it work on these corpora?**
  - BioScope (EN) [Vincze et al., 2009]
    - 3 sub-corpora (Abstract, Full, Clinical)
  - SFUPProductReview (EN) [Konstantinova et al., 2012]
  - CNeSp (ZH) [Zou et al., 2015]
    - 3 sub-corpora (Product, Financial, Scientific)
Joint model

- Same bi-LSTM architecture, same features
- Add a 4-parameter transition matrix to create the dependency on the previous output

\[ p(s|w, c) = \prod_{i=1}^{n} p(s_i|s_{i-1}, w, c) \]
Evaluation

- Token-level: $F_1$ on tokens correctly classified
- Scope-level: Accuracy of full scopes we correctly match
- Performance on par or better than previous work
Rule-based scope detection

A lot of sentences where scope is delimited by punctuation

It helps activation, not inhibition of ibrf1 cells.
Results

Token-level $F_1$

<table>
<thead>
<tr>
<th>Sherlock</th>
<th>SFU</th>
<th>BioScope Abstract</th>
<th>BioScope Full</th>
<th>BioScope Clinical</th>
<th>CNeSp Product</th>
<th>CNeSp Financial</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Rule-based</td>
<td>joint</td>
<td>Rule-based</td>
<td>joint</td>
<td>Rule-based</td>
</tr>
</tbody>
</table>

- Rule-based
- joint
Results

Scope-level accuracy

- Sherlock
- SFU
- BioScope Abstract
- BioScope Full
- BioScope Clinical
- CNeSp Product
- CNeSp Financial

- Rule-based
- joint

0 20 40 60 80 100

Scores for different tools and scopes.
Blame it on the training data

It helps activation, not inhibition of ibrf1 cells.

avg. = 65
Easy vs. hard instances

- **Easy**: predictable by punctuation
  - It helps activation, *not inhibition* of ibrf1 cells.

- **Hard**: not predictable by punctuation
  - I do *not use* the 56k conextant winmodem since I have cable access for the internet and he does not either.
Error analysis: dev set

- Sherlock
- SFU
- BioScope Abstract
- BioScope Full
- BioScope Clinical
- CNeSp Product
- CNeSp Financial

% easy correct % hard correct

- [Bar chart showing performance for each category]
Error analysis: *dev set*

- Most of the errors are due to the model trying to match punctuation boundaries

  surprisingly, expression of *neither bhrf1 nor blc-2* in a *b-cell line bjab*, protected by the cells from anti-fas-mediated apostosis
Error analysis: dev set

- Most of the errors are due to the model trying to match punctuation boundaries

surprisingly, expression of neither bhrf1 nor blc-2 in a b-cell line bjab, protected by the cells from anti-fas-mediated apostosis
Error analysis: dev set

- Most of the errors are due to the model trying to match punctuation boundaries

surprisingly, expression of neither bhrf1 nor blc-2 in a b-cell line

bjab, protected by the cells from anti-fas-mediated apoptosis

I do not use the 56k conextant winmodem since I have cable
access for the internet.
Error analysis: dev set

- Most of the errors are due to the model trying to match punctuation boundaries

  surprisingly, expression of neither bhrf1 nor blc-2 in a b-cell line bjab, protected by the cells from anti-fas-mediated apoptosis

  I do not use the 56k conextant winmodem since I have cable access for the internet.
Why does it happen?

Different corpora, different annotation styles

| BioScope & SFU |  
|----------------|---
| CNeSp          |  
| Sherlock       |  
Why does it happen?

Different corpora, different annotation styles

<table>
<thead>
<tr>
<th></th>
<th>It helps activation, not inhibition of ibrf1 cells.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioScope &amp; SFU</td>
<td></td>
</tr>
<tr>
<td>CNeSp</td>
<td></td>
</tr>
<tr>
<td>Sherlock</td>
<td></td>
</tr>
</tbody>
</table>

Subject is seldom annotated
Why does it happen?

Different corpora, different annotation styles

<table>
<thead>
<tr>
<th>Corpora</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioScope &amp; SFU</td>
<td>It helps activation, <strong>not inhibition</strong> of ibrf1 cells.</td>
</tr>
<tr>
<td>CNeSp</td>
<td>It helps activation, <strong>not inhibition</strong> of ibrf1 cells.</td>
</tr>
<tr>
<td>Sherlock</td>
<td><strong>It helps</strong> activation, <strong>not inhibition</strong> of ibrf1 cells.</td>
</tr>
</tbody>
</table>

**Subject** is always annotated, **omitted verb** is retrieved
Is this problem caused by the annotation guidelines?

- We re-annotated 100 randomly selected sentences of 3 corpora using the Sherlock guidelines.

<table>
<thead>
<tr>
<th>Data</th>
<th>Easy original</th>
<th>Easy Sherlock</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFU</td>
<td>87%</td>
<td>42%</td>
</tr>
<tr>
<td>BioScope Abstract</td>
<td>84%</td>
<td>34%</td>
</tr>
<tr>
<td>CNeSp Financial</td>
<td>68%</td>
<td>45%</td>
</tr>
</tbody>
</table>
Undersampling is not enough

![Graph showing the relationship between the percentage of punctuated instances in training and accuracy. The graph includes lines for 'punct dev', 'punct tst', 'no punct dev', and 'no punct tst'. The x-axis represents the percentage of punctuated instances in training, ranging from 0% to 100%, and the y-axis represents accuracy, ranging from 0% to 100%. The graph shows that increasing the percentage of punctuated instances in training leads to an increase in accuracy for both dev and test sets, but the 'no punct' cases remain relatively flat.]
Conclusions

- GOOD PERFORMANCE FEELS GREAT BUT UNDERSTANDING YOUR MODEL FEELS EVEN BETTER!
- Detecting negation scope is easy, except when it isn’t:
  - focus detection on those more difficult cases?