Density-Driven Cross-Lingual Transfer of Dependency Parsers

Mohammad Sadegh Rasooli    Michael Collins
rasooli@cs.columbia.edu

Presented by
Owen Rambow

EMNLP 2015
Motivation

Availability of treebanks

- Accurate parsers use annotated treebanks.
- There are no gold-standard treebanks for many languages.
- Annotated treebanks are very expensive to create.

Common approach: using universal linguistic information

- Without parallel data; e.g [Zhang and Barzilay, 2015]
- With parallel data; e.g [Ma and Xia, 2014]
  - The best results but still lags behind supervised parsing
Motivation

Availability of treebanks
- Accurate parsers use annotated treebanks.
- There are no gold-standard treebanks for many languages.
- Annotated treebanks are very expensive to create.

Common approach: using universal linguistic information
- Without parallel data; e.g. [Zhang and Barzilay, 2015]
- With parallel data; e.g. [Ma and Xia, 2014]
  - The best results but still lags behind supervised parsing
The political priorities must be set by this House and the MEPs.

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
The political priorities must be set by this House and the MEPs.

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
Parse

Parse source sentence with a supervised parser.
Projecting Dependencies from Parallel Data

Project dependencies.

The political priorities must be set by this House and the MEPs.

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
Train on the projected dependencies.

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden.
Practical Problems

- Most translations are not word-to-word.
- Alignment errors!
- Supervised parsers are not perfect.
- Difference in syntactic behavior across languages.
Previous Results

dependency accuracy (avg. over 6 EU languages)

- [McDonald et al., 2011]
- [Zhang and Barzilay, 2015]
- [Ma and Xia, 2014]

Previous work

- MPH11: 71.34
- ZB15: 75.4
- MX14: 76.67
Previous work
Supervised models

8% lower than a first-order supervised model.

dependency accuracy (avg. over 6 EU languages)
Our Approach

- We define different sets of dense structures
  - Full trees
  - Dense partial trees
A projected **full tree** $t \in P_{100}$ is:

- A projective dependency tree
- All words have one parent
A partial tree \( t \in P_{80} \) is:

- A projective dependency tree (a collection of projective trees)
- At least 80% of words have one parent
A partial tree $t \in \mathcal{P}_{\geq k}$ is:

- A projective dependency tree (a collection of projective trees)
- There is at least one span of length $\geq k$ where all words in that span have one parent

$k=7$ in the above tree
Our Contributions

- We demonstrate the utility of dense projected structures.
- We describe a training algorithm that builds on dense structures.
Our Contributions

Previous work

Supervised models

dependency accuracy (avg. over 6 EU languages)

- MPH11: 71.34
- ZB15: 75.4
- MX14: 76.67
- Our Model: 84.29
- 1st-ord: 87.5
- Sh-R: 87.5

[McDonald et al., 2011]
[Zhang and Barzilay, 2015]
[Ma and Xia, 2014]
[McDonald et al., 2005]
[Rasooli and Tetreault, 2015]

Mohammad Sadegh Rasooli, Michael Collins
Our Contributions

- Our Model

Comparison of dependency accuracy (avg. over 6 EU languages):

- MPH11: 71.34
- ZB15: 75.4
- MX14: 76.67
- Our Model: 82.18
- 1st-ord: 84.29
- Sh-R: 87.5

5.5% absolute improvement

Previous work

- Supervised models

- [McDonald et al., 2011]
- [Zhang and Barzilay, 2015]
- [Ma and Xia, 2014]
- [McDonald et al., 2005]
- [Rasooli and Tetreault, 2015]
Overview

- The learning algorithm
- Results
- Analysis
Languages from Google universal treebank:
- English (only as source), German, Spanish, French, Italian, Portuguese, and Swedish.
- English to German transfer data for developing our models.

We use Giza++ intersected alignments on EuroParl data.

We use the Yara parser [Rasooli and Tetreault, 2015], a shift-reduce beam parser.
We use the following function definitions:

- \text{Train}(D)
- \text{CDECODE}(P, \theta)
- \text{TOP}(D, \theta)
Train($D$)

- Input $D$
  - A set of dependency trees (full trees)
- Output $\theta$
  - A parsing model
CDECODE($P, \theta$)

- Input $P$
  - A set of partial dependency structures
- Input $\theta$
  - Parsing model
- Output $D$
  - A set of full trees that are completely consistent with the dependencies in $P$.
  - Filling in partial trees with dynamic oracles
    [Goldberg and Nivre, 2013].

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden. $ROOT$
CDECODE($P, \theta$)

- Input $P$
  - A set of partial dependency structures
- Input $\theta$
  - Parsing model
- Output $D$
  - A set of full trees that are completely consistent with the dependencies in $P$
  - Filling in partial trees with dynamic oracles [Goldberg and Nivre, 2013].

Die politischen Prioritäten müssen von diesem Parlament und den Europaabgeordneten abgesteckt werden. 

Mohammad Sadegh Rasooli, Michael Collins  
Density-Driven Cross-Lingual Transfer of Dependency Parsers
TOP($D, \theta$)

- **Input $D$**
  - A set of full dependency trees
- **Input $\theta$**
  - Parsing model
- **Output $A$**
  - Top $m$ highest scoring trees in $D$
    - We use $m=200,000$ in our experiments.
  - **Score**: Perceptron-based parse score normalized by sentence length
Definitions

- $A_0 = P_{100}$
- $A_1 = P_{\geq 7} \cup P_{80}$
- $A_2 = P_{\geq 5} \cup P_{80}$
- $A_3 = P_{\geq 1} \cup P_{80}$

Note $A_1 \subseteq A_2 \subseteq A_3$
Learning Algorithm

Train on full trees

\[ \theta_0 = \text{Train}(A_0) \]

\textbf{for} \( i = 1 \ldots 3 \) \textbf{do}

\[ D_i = \text{CDECODE}(A_i, \theta_{i-1}) \]
\[ A'_i = \text{TOP}(D_i, \theta_{i-1}) \]
\[ \theta_i = \text{Train}(A_0 \cup A'_i) \]

\textbf{end for}

Return \( \theta_3 \)

Given definitions:

- \( A_0 = P_{100} \)
- \( A_1 = P_{\geq 7} \cup P_{80} \)
- \( A_2 = P_{\geq 5} \cup P_{80} \)
- \( A_3 = P_{\geq 1} \cup P_{80} \)

Note \( A_1 \subseteq A_2 \subseteq A_3 \)
Gradually decrease density

\[ \theta_0 = \text{Train}(A_0) \]

\textbf{for} i = 1 \ldots 3 \textbf{do}
\[ D_i = \text{CDECODE}(A_i, \theta_{i-1}) \]
\[ A'_i = \text{TOP}(D_i, \theta_{i-1}) \]
\[ \theta_i = \text{Train}(A_0 \cup A'_i) \]
\textbf{end for}

Return \( \theta_3 \)

Given definitions:

- \( A_0 = P_{100} \)
- \( A_1 = P_{\geq 7} \cup P_{80} \)
- \( A_2 = P_{\geq 5} \cup P_{80} \)
- \( A_3 = P_{\geq 1} \cup P_{80} \)

Note \( A_1 \subseteq A_2 \subseteq A_3 \)
Learning Algorithm

Fill in partial trees

\[ \theta_0 = \text{Train}(A_0) \]

\textbf{for} \( i = 1 \ldots 3 \) \textbf{do}

\[ D_i = \text{CDECODE}(A_i, \theta_{i-1}) \]

\[ A'_i = \text{TOP}(D_i, \theta_{i-1}) \]

\[ \theta_i = \text{Train}(A_0 \cup A'_i) \]

\textbf{end for}

Return \( \theta_3 \)

Given definitions:

- \( A_0 = P_{100} \)
- \( A_1 = P_{\geq 7} \cup P_{80} \)
- \( A_2 = P_{\geq 5} \cup P_{80} \)
- \( A_3 = P_{\geq 1} \cup P_{80} \)

Note \( A_1 \subseteq A_2 \subseteq A_3 \)
Learning Algorithm

Select high-scoring trees

\[ \theta_0 = \text{Train}(A_0) \]

\textbf{for} \; i = 1 \ldots 3 \; \textbf{do}

\[ D_i = \text{CDECODE}(A_i, \theta_{i-1}) \]

\[ A'_i = \text{TOP}(D_i, \theta_{i-1}) \]

\[ \theta_i = \text{Train}(A_0 \cup A'_i) \]

\textbf{end for}

Return \( \theta_3 \)

Given definitions:

- \( A_0 = P_{100} \)
- \( A_1 = P_{\geq 7} \cup P_{80} \)
- \( A_2 = P_{\geq 5} \cup P_{80} \)
- \( A_3 = P_{\geq 1} \cup P_{80} \)

Note \( A_1 \subseteq A_2 \subseteq A_3 \)
Learning Algorithm

Train on the new set

\[ \theta_0 = \text{Train}(A_0) \]

for \( i = 1 \ldots 3 \) do

\[ D_i = \text{CDECODE}(A_i, \theta_{i-1}) \]
\[ A'_i = \text{TOP}(D_i, \theta_{i-1}) \]
\[ \theta_i = \text{Train}(A_0 \cup A'_i) \]

end for

Return \( \theta_3 \)

Given definitions:

- \( A_0 = P_{100} \)
- \( A_1 = P_{\geq 7} \cup P_{80} \)
- \( A_2 = P_{\geq 5} \cup P_{80} \)
- \( A_3 = P_{\geq 1} \cup P_{80} \)

Note \( A_1 \subseteq A_2 \subseteq A_3 \)
Learning Algorithm

Return the final model

\[ \theta_0 = \text{Train}(A_0) \]

\textbf{for} \( i = 1 \ldots 3 \) \textbf{do}

\[ D_i = \text{CDECODE}(A_i, \theta_{i-1}) \]
\[ A'_i = \text{TOP}(D_i, \theta_{i-1}) \]
\[ \theta_i = \text{Train}(A_0 \cup A'_i) \]

\textbf{end for}

Return \( \theta_3 \)

Given definitions:

- \( A_0 = P_{100} \)
- \( A_1 = P_{\geq 7} \cup P_{80} \)
- \( A_2 = P_{\geq 5} \cup P_{80} \)
- \( A_3 = P_{\geq 1} \cup P_{80} \)

Note \( A_1 \subseteq A_2 \subseteq A_3 \)
Overview

- The learning algorithm
- Results
- Analysis
Two Settings

Scenario 1
- Transfer from English.

Scenario 2 (voting)
- The different languages vote on dependencies.
  - This scenario is true for cases such as Europarl.
Results on European Languages

Density-Driven Cross-Lingual Transfer of Dependency Parsers

Mohammad Sadegh Rasooli, Michael Collins
Results on European Languages

Mohammad Sadegh Rasooli, Michael Collins

Density-Driven Cross-Lingual Transfer of Dependency Parsers
Results on European Languages

German
Spanish
French
Italian
Portuguese
Swedish

UAS

θ₀ (Full trees)  θ₃ (Full+dense)  θ₃ (voting)
Results on European Languages (Comparison)

<table>
<thead>
<tr>
<th>Language</th>
<th>Ma and Xia, 2014</th>
<th>$\theta_3$ (voting)</th>
<th>Sup. MST-1st</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>74.3</td>
<td>79.68</td>
<td>81.65</td>
</tr>
<tr>
<td>Spanish</td>
<td>76.53</td>
<td>80.86</td>
<td>83.92</td>
</tr>
<tr>
<td>French</td>
<td>76.53</td>
<td>82.72</td>
<td>83.51</td>
</tr>
<tr>
<td>Italian</td>
<td>77.74</td>
<td>83.67</td>
<td>85.47</td>
</tr>
<tr>
<td>Portuguese</td>
<td>76.65</td>
<td>82.07</td>
<td>85.67</td>
</tr>
<tr>
<td>Swedish</td>
<td>79.27</td>
<td>84.06</td>
<td>85.59</td>
</tr>
</tbody>
</table>
Comparison to Previous Work

Previous work

\[\text{dependency accuracy (avg. over 6 EU languages)}\]

- MPH11
- ZB15
- MX14

\[\theta_0\]
\[\theta_3\] (voting)

Supervised models

\[\text{1st-ord}\]
\[\text{Sh-R}\]

- [McDonald et al., 2011]
- [Zhang and Barzilay, 2015]
- [Ma and Xia, 2014]
- [McDonald et al., 2005]
- [Rasooli and Tetreault, 2015]
**Comparison to Previous Work**

<table>
<thead>
<tr>
<th>Model</th>
<th>Dependency Accuracy (avg. over 6 EU languages)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPH11</td>
<td>71.34</td>
</tr>
<tr>
<td>ZB15</td>
<td>75.4</td>
</tr>
<tr>
<td>MX14</td>
<td>76.67</td>
</tr>
<tr>
<td>(\theta_0) (full trees)</td>
<td>75.88</td>
</tr>
<tr>
<td>(\theta_3) (full+dense)</td>
<td>78.89</td>
</tr>
<tr>
<td>[McDonald et al., 2005]</td>
<td>84.29</td>
</tr>
<tr>
<td>[Rasooli and Tetreault, 2015]</td>
<td>87.5</td>
</tr>
</tbody>
</table>

2.2% absolute improvement

**Previous work**

**Supervised models**

Mohammad Sadegh Rasooli, Michael Collins

Density-Driven Cross-Lingual Transfer of Dependency Parsers
Comparison to Previous Work

- [McDonald et al., 2011]
- [Zhang and Barzilay, 2015]
- [Ma and Xia, 2014]
- [McDonald et al., 2005]
- [Rasooli and Tetreault, 2015]

Dependency accuracy (avg. over 6 EU languages): 71.34, 75.4, 76.67, 75.88, 78.89, 82.18, 84.29, 87.5

5.5% absolute improvement

Supervised models

Previous work
Overview

- The learning algorithm
- Results
- **Analysis**
The accuracy of full trees is high.

Voting increases the number of words per sentence, number of sentences and accuracy of full trees.

<table>
<thead>
<tr>
<th>Setting</th>
<th>English→target</th>
<th>Voting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sen#</td>
<td>17K</td>
<td>77K</td>
</tr>
<tr>
<td>Word/sen</td>
<td>6.8</td>
<td>10.4</td>
</tr>
<tr>
<td>Prec. vs supervised</td>
<td>84.7</td>
<td>89.0</td>
</tr>
</tbody>
</table>
The length and number of sentences are increased in partial dense trees.
The accuracy of partial trees are lower than full trees.

<table>
<thead>
<tr>
<th>Setting</th>
<th>$P_{100}$</th>
<th>$P_{80} \cup P_{\geq 7}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sen#</td>
<td>77K</td>
<td>243K</td>
</tr>
<tr>
<td>Deps#</td>
<td>10.4</td>
<td>13.7</td>
</tr>
<tr>
<td>Words/sen</td>
<td>10.4</td>
<td>27.6</td>
</tr>
<tr>
<td>Density</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>Prec. vs supervised</td>
<td>89.0</td>
<td>84.7</td>
</tr>
</tbody>
</table>
# Accuracy across Different Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>$P_{100}$</th>
<th></th>
<th></th>
<th>$P_{80} \cup P_{\geq 7}$</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#sen</td>
<td>words/sen</td>
<td>#dep</td>
<td>Prec.</td>
<td>#sen</td>
<td>words/sen</td>
<td>#dep</td>
</tr>
<tr>
<td>German</td>
<td>47K</td>
<td>8.2</td>
<td>8.2</td>
<td>91.4</td>
<td>75K</td>
<td>23.5</td>
<td>10.8</td>
</tr>
<tr>
<td>Spanish</td>
<td>109K</td>
<td>12.1</td>
<td>12.1</td>
<td>89.2</td>
<td>346K</td>
<td>28.5</td>
<td>17.0</td>
</tr>
<tr>
<td>French</td>
<td>78K</td>
<td>11.7</td>
<td>11.7</td>
<td>91.2</td>
<td>303K</td>
<td>29.9</td>
<td>14.9</td>
</tr>
<tr>
<td>Italian</td>
<td>101K</td>
<td>12.4</td>
<td>12.4</td>
<td>87.9</td>
<td>301K</td>
<td>28.5</td>
<td>15.2</td>
</tr>
<tr>
<td>Portuguese</td>
<td>39K</td>
<td>8.8</td>
<td>8.8</td>
<td>85.8</td>
<td>222K</td>
<td>30.3</td>
<td>12.4</td>
</tr>
<tr>
<td>Swedish</td>
<td>86K</td>
<td>9.5</td>
<td>9.5</td>
<td>88.8</td>
<td>211K</td>
<td>25.2</td>
<td>12.2</td>
</tr>
<tr>
<td>Average</td>
<td>77K</td>
<td>10.4</td>
<td>10.4</td>
<td>89.0</td>
<td>243K</td>
<td>27.6</td>
<td>13.7</td>
</tr>
</tbody>
</table>

Mohammad Sadegh Rasooli, Michael Collins

Density-Driven Cross-Lingual Transfer of Dependency Parsers
Conclusion

- We showed the utility of dense structures in projected dependencies.
- We showed a simple and effective learning method to utilize dense structures.
- Our performance is very close to a supervised parser.
- Future work:
  - Applying to a broader set of languages.
  - Using this model to improve machine translation.
Thanks

Bloomberg


