Motivation
- Existing parsers fail to capture long range structural dependencies between constituents of a discourse tree.
- A reranker can exploit the global information as follows:
  1. A base parser produces k hypotheses.
  2. A classifier selects the best hypothesis by exploiting entire information in each hypothesis.
- Tree Kernels (TKs) allow kernel-based learning models like SVMs to learn from arbitrary tree fragments.

Our Method
- Extend the parser of Joty et. al., (2013) to k-best parsing.
- Define novel kernels for discourse trees based on new representations.
- Use SVM preference reranking framework to rank k hypotheses and select the best tree.

Experiments (RST-DT)
- 385 news articles
  - Train: 347 (7321 sent.)
  - Test: 38 (951 sent.)
  - Relation set: 18 coarser
- 5-fold CV was used to generate reranking data

Oracle scores
- Sentence-level
- Document-level

Results
- Which TK works better on which representation?

<table>
<thead>
<tr>
<th>TK</th>
<th>JRN</th>
<th>SRN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>81.28</td>
<td>82.15</td>
</tr>
<tr>
<td>All</td>
<td>80.04</td>
<td>80.04</td>
</tr>
</tbody>
</table>

- How does reranking performance vary for different values of k?

<table>
<thead>
<tr>
<th>k</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAR-D</td>
<td>79.77</td>
<td>84.42</td>
<td>86.55</td>
<td>87.68</td>
<td>88.09</td>
<td>88.73</td>
<td>90.74</td>
<td>92.57</td>
<td>92.90</td>
<td>93.22</td>
</tr>
<tr>
<td>STK</td>
<td>81.28</td>
<td>80.04</td>
<td>82.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STKk</td>
<td>81.35</td>
<td>80.28</td>
<td>82.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTK</td>
<td>81.63</td>
<td>78.50</td>
<td>81.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Which features are important?

<table>
<thead>
<tr>
<th>Feature</th>
<th>+ Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>79.77</td>
</tr>
<tr>
<td>Basic feat.</td>
<td>79.84</td>
</tr>
<tr>
<td>+ Rel. feat.</td>
<td>79.81</td>
</tr>
<tr>
<td>+ Tree</td>
<td>82.15</td>
</tr>
</tbody>
</table>

Preference Reranking
- Given a pair of hypotheses \( h_i, h_j \), a classifier decides whether \( h_i \) is a better tree than \( h_j \).
- Training: +ve \( h_i, h_j \) -ve \( h_j, h_i \)
  \( h_i \) has the highest f-score accuracy
- Testing: Apply the classifier to all possible pairs and take votes to rank the k candidates.

Preference Kernel

\[
PK(h_1, h_2) = \frac{1}{K(h_1, h_2) + K(h_2, h_1)}
\]

Non sub-tree features
- Base parser rank & probability
- Structural properties of the DT
- Relation features

Our Findings
- Bigram lexicalization is better then All.
- STK performs better than PTK on SRN.
- Best result is obtained for k=4.5 on std. testset.
- Improvement is consistent on whole corpus.
- Best result is obtained for k=6 on whole corpus.
- Non-subtree features doesn’t help much.
- Subtree features learnt automatically are indeed crucial for the performance gain.
- Reranking at the sentence-level significantly pushes the state-of-the-art overall accuracy.

Reference: