Annotation Schemes and Parser Evaluation for German

Ines Rehbein and Josef van Genabith

NCLT, Dublin City University
Outline

1 Motivation

2 Experiments
   - Experimental Setup
   - Experiment I: Controlled Error Insertion
   - Experiment II: Cross-Treebank Conversion
   - Experiment III: Dependency-Based Evaluation

3 Conclusions
Research Questions

- Is parsing non-configurational languages like German harder than parsing English?
- What is the impact of different treebank annotation schemes on PCFG parsing results?
- How can we compare parser output for parsers trained on treebanks with different annotation schemes?
  - Is PARSEVAL a reliable measure to assess parse quality?
  - Can PARSEVAL be used to compare parsers trained on different treebanks?
Research Questions

- Is parsing non-configurational languages like German harder than parsing English?
- What is the impact of different treebank annotation schemes on PCFG parsing results?
- How can we compare parser output for parsers trained on treebanks with different annotation schemes?
  - Is PARSEVAL a reliable measure to assess parse quality?
  - Can PARSEVAL be used to compare parsers trained on different treebanks?
Research Questions

- Is parsing non-configurational languages like German harder than parsing English?
- What is the impact of different treebank annotation schemes on PCFG parsing results?
- How can we compare parser output for parsers trained on treebanks with different annotation schemes?
- Is PARSEVAL a reliable measure to assess parse quality?
- Can PARSEVAL be used to compare parsers trained on different treebanks?
Research Questions

- Is parsing non-configurational languages like German harder than parsing English?
- What is the impact of different treebank annotation schemes on PCFG parsing results?
- How can we compare parser output for parsers trained on treebanks with different annotation schemes?
  - Is PARSEVAL a reliable measure to assess parse quality?
  - Can PARSEVAL be used to compare parsers trained on different treebanks?
The two Treebanks: TIGER and TüBa-D/Z

“IT also looks bad for public contracts.”

“Things seem better in Wales.”

Ines Rehbein and Josef van Genabith
Annotation Schemes and Parser Evaluation for German
Parser Evaluation along different Dimensions

- **PARSEVAL (tree-based)**
  - checks label and wordspan identity
  - does not weight results
  - does not give credit to constituents with correct label but slightly wrong phrase boundary

- **Leaf-Ancestor (string-based)**
  - measures path similarity from terminal node to root node
  - computes Levenshtein distance for each path

- **Dependency-based Evaluation**
  - maps CFG trees to dependency relations
Parser Evaluation along different Dimensions

- **PARSEVAL (tree-based)**
  - checks label and wordspan identity
  - does not weight results
  - does not give credit to constituents with correct label but slightly wrong phrase boundary

- **Leaf-Ancestor (string-based)**
  - measures path similarity from terminal node to root node
  - computes Levenshtein distance for each path

- Dependency-based Evaluation
  - maps CFG trees to dependency relations
Parser Evaluation along different Dimensions

- PARSEVAL (tree-based)
  - checks label and wordspan identity
  - does not weight results
  - does not give credit to constituents with correct label but slightly wrong phrase boundary

- Leaf-Ancestor (string-based)
  - measures path similarity from terminal node to root node
  - computes Levenshtein distance for each path

- Dependency-based Evaluation
  - maps CFG trees to dependency relations
Experiment I: Error Insertion

Test set:
- 1024 sentences from each treebank
- comparable with regard to word length, syntactic structure and complexity

Create the test set:
- select sentences from each treebank with sentence length $10 \leq n \leq 40$
- for all sentences with equal length compute average number of prep, det, nouns, ...
- for sentence length $n$: select all sentences where average for each POS does not deviate more than 0.8 from average for all sentences with length $n$
Experiment I: Error Insertion

- Test set:
  - 1024 sentences from each treebank
  - comparable with regard to word length, syntactic structure and complexity

- Create the test set:
  - select sentences from each treebank with sentence length $10 \leq n \leq 40$
  - for all sentences with equal length compute average number of prep, det, nouns, ...
  - for sentence length $n$: select all sentences where average for each POS does not deviate more than 0.8 from average for all sentences with length $n$
Experiment I: Error Insertion

- Controlled error insertion into gold treebank trees
- 3 error types: Attachment, Label and Span Errors

<table>
<thead>
<tr>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTACH I</td>
</tr>
<tr>
<td>ATTACH II</td>
</tr>
<tr>
<td>LABEL I</td>
</tr>
<tr>
<td>LABEL II</td>
</tr>
<tr>
<td>SPAN I</td>
</tr>
<tr>
<td>SPAN II</td>
</tr>
<tr>
<td>SPAN III</td>
</tr>
</tbody>
</table>

- ATTACH I: Attach PPs inside an NP one level higher up in the tree
- ATTACH II: Change verb attachment to noun attachment for PPs (S, VP, MF)
- LABEL I: Change labels of PPs to NP
- LABEL II: Change labels of VPs to PP
- SPAN I: Include adverb to the left of a PP into the PP
- SPAN II: Include NN to the left of a PP into the PP
- SPAN III: Combination of SPAN I and SPAN II
### Experiment I: Results

<table>
<thead>
<tr>
<th></th>
<th>PARSEVAL</th>
<th></th>
<th>Leaf-Ancestor</th>
<th></th>
<th># errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TIGER</td>
<td>TüBa</td>
<td>TIGER</td>
<td>TüBa</td>
<td></td>
</tr>
<tr>
<td>PP attachment I</td>
<td>98.84</td>
<td>99.57</td>
<td>99.62</td>
<td>99.70</td>
<td>85</td>
</tr>
<tr>
<td>PP attachment II</td>
<td>98.75</td>
<td>99.55</td>
<td>99.66</td>
<td>99.78</td>
<td>89</td>
</tr>
<tr>
<td>Label I</td>
<td>80.02</td>
<td>92.73</td>
<td>92.45</td>
<td>95.24</td>
<td>1427</td>
</tr>
<tr>
<td>Label II</td>
<td>93.00</td>
<td>97.45</td>
<td>96.05</td>
<td>99.28</td>
<td>500</td>
</tr>
<tr>
<td>SPAN I</td>
<td>99.01</td>
<td>99.64</td>
<td>99.82</td>
<td>99.84</td>
<td>71</td>
</tr>
<tr>
<td>SPAN II</td>
<td>97.47</td>
<td>99.08</td>
<td>99.51</td>
<td>99.77</td>
<td>181</td>
</tr>
<tr>
<td>SPAN III</td>
<td>96.51</td>
<td>98.73</td>
<td>99.34</td>
<td>99.62</td>
<td>252</td>
</tr>
<tr>
<td>total weighted ave.</td>
<td>87.09</td>
<td>97.18</td>
<td>94.98</td>
<td>97.18</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Results for controlled error insertion into gold treebank trees
Die Stadtverwaltung von Venedig hat erstmals streunende Katzen gezählt.
“For the first time the city council of Venice has counted straying cats.”

(S
  (NP Die Stadtverwaltung [the city counsel])
  (PP von Venedig [of Venice])
  hat [has]
  erstmals [for the first time]
  (VP
    (NP streunende Katzen [straying cats])
    gezählt [counted]
  )
)
Comparison: PARSEVAL - LA (II)

<table>
<thead>
<tr>
<th>ERROR</th>
<th>PARSEVAL</th>
<th>LA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP attachment I</td>
<td>83.33</td>
<td>96.30</td>
</tr>
<tr>
<td>Label I</td>
<td>83.33</td>
<td>96.00</td>
</tr>
<tr>
<td>Label II</td>
<td>83.33</td>
<td>91.00</td>
</tr>
<tr>
<td>SPAN II</td>
<td>83.33</td>
<td>96.40</td>
</tr>
</tbody>
</table>

- LA results are sensitive to the number of terminal nodes affected by the error.
- LA favours TüBa-D/Z because of the higher ratio of non-terminal/terminal nodes.
- PARSEVAL highly favours annotation schemes with a higher ratio of non-terminal/terminal nodes.
## Comparison: PARSEVAL - LA (II)

<table>
<thead>
<tr>
<th>ERROR</th>
<th>PARSEVAL</th>
<th>LA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP attachment I</td>
<td>83.33</td>
<td>96.30</td>
</tr>
<tr>
<td>Label I</td>
<td>83.33</td>
<td>96.00</td>
</tr>
<tr>
<td>Label II</td>
<td>83.33</td>
<td>91.00</td>
</tr>
<tr>
<td>SPAN II</td>
<td>83.33</td>
<td>96.40</td>
</tr>
</tbody>
</table>

- LA results are sensitive to the number of terminal nodes affected by the error
- LA favours TüBa-D/Z because of the higher ratio of non-terminal/terminal nodes
- PARSEVAL highly favours annotation schemes with a higher ratio of non-terminal/terminal nodes
Comparison: PARSEVAL - LA (II)

<table>
<thead>
<tr>
<th>ERROR</th>
<th>PARSEVAL</th>
<th>LA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP attachment I</td>
<td>83.33</td>
<td>96.30</td>
</tr>
<tr>
<td>Label I</td>
<td>83.33</td>
<td>96.00</td>
</tr>
<tr>
<td>Label II</td>
<td>83.33</td>
<td>91.00</td>
</tr>
<tr>
<td>SPAN II</td>
<td>83.33</td>
<td>96.40</td>
</tr>
</tbody>
</table>

- LA results are sensitive to the number of terminal nodes affected by the error
- LA favours TüBa-D/Z because of the higher ratio of non-terminal/terminal nodes
- PARSEVAL highly favours annotation schemes with a higher ratio of non-terminal/terminal nodes
Comparison: PARSEVAL - LA (II)

<table>
<thead>
<tr>
<th>ERROR</th>
<th>PARSEVAL</th>
<th>LA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP attachment I</td>
<td>83.33</td>
<td>96.30</td>
</tr>
<tr>
<td>Label I</td>
<td>83.33</td>
<td>96.00</td>
</tr>
<tr>
<td>Label II</td>
<td>83.33</td>
<td>91.00</td>
</tr>
<tr>
<td>SPAN II</td>
<td>83.33</td>
<td>96.40</td>
</tr>
</tbody>
</table>

- LA results are sensitive to the number of terminal nodes affected by the error
- LA favours TüBa-D/Z because of the higher ratio of non-terminal/terminal nodes
- PARSEVAL highly favours annotation schemes with a higher ratio of non-terminal/terminal nodes
Kübler (2005), Maier (2006):
- Converted TüBa-D/Z trees to a TIGER-like format
- Removed unary nodes, topological fields, flattened phrases
- After modifying the original treebank they extracted a PCFG
- All modifications lead to an approximation of the PARSEVAL results to the results of the other treebank

We convert the parser output of the TüBa-D/Z to a TIGER-like format

Converted trees encode the same basic syntactic structure and same errors
→ Results should not be crucially different!
Experiment II: Treebank Conversion

- Kübler (2005), Maier (2006):
  - Converted TüBa-D/Z trees to a TIGER-like format
  - Removed unary nodes, topological fields, flattened phrases
  - After modifying the original treebank they extracted a PCFG
  - All modifications lead to an approximation of the PARSEVAL results to the results of the other treebank

- We convert the parser output of the TüBa-D/Z to a TIGER-like format
  - Converted trees encode the same basic syntactic structure and same errors
    → Results should not be crucially different!
**Experiment II: Treebank Conversion**

- Kübler (2005), Maier (2006):
  - Converted TüBa-D/Z trees to a TIGER-like format
  - Removed unary nodes, topological fields, flattened phrases
  - After modifying the original treebank they extracted a PCFG
  - All modifications lead to an approximation of the PARSEVAL results to the results of the other treebank

- We convert the parser output of the TüBa-D/Z to a TIGER-like format
- Converted trees encode the same basic syntactic structure and same errors
  - Results should not be crucially different!
Erziehungsurlaub nehmen bisher nur zwei Prozent der Männer.
"Until now only two percent of the men take parental leave."

Figure: Original TüBa-D/Z-style gold tree
Figure: Converted TIGER-style gold tree
Experiment II: Treebank Conversion

**Figure:** Parser output (TüBa-D/Z grammar)
Experiment II: Treebank Conversion

Figure: Converted parser output (TüBa-D/Z)
## Experiment II: Results

<table>
<thead>
<tr>
<th></th>
<th>PARSEVAL</th>
<th></th>
<th>LA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>prec.</td>
<td>recall</td>
<td>F-sco.</td>
</tr>
<tr>
<td><strong>TIGER</strong></td>
<td>83.54</td>
<td>83.65</td>
<td>83.59</td>
</tr>
<tr>
<td>no Unary</td>
<td>84.33</td>
<td>84.48</td>
<td>84.41</td>
</tr>
<tr>
<td><strong>TüBa-D/Z</strong></td>
<td>92.59</td>
<td>89.79</td>
<td>91.17</td>
</tr>
<tr>
<td><strong>TüBa-D/Z → TIGER</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no Top</td>
<td>92.38</td>
<td>88.76</td>
<td>90.53</td>
</tr>
<tr>
<td>no Unary</td>
<td>89.96</td>
<td>85.67</td>
<td>87.76</td>
</tr>
<tr>
<td>no Top + no U.</td>
<td>88.44</td>
<td>82.24</td>
<td>85.23</td>
</tr>
<tr>
<td>no Top + no U. + no NP in PP</td>
<td>87.15</td>
<td>79.52</td>
<td>83.16</td>
</tr>
</tbody>
</table>

**Table:** The impact of the conversion process on PARSEVAL and LA
Experiment II: Conclusions

- Constant decrease in PARSEVAL results for modified trees is consistent with results in Kübler (2005) and Maier (2006)

- But: no change in quality for parser output, only change in encoding

  \[ same \] parser output with TIGER-style encoding yields lower PARSEVAL results

  - TüBa-D/Z grammar yields higher precision in the PARSEVAL metric
  - TIGER yields higher recall for PARSEVAL
Experiment II: Conclusions

- Constant decrease in PARSEVAL results for modified trees is consistent with results in Kübler (2005) and Maier (2006)
- But: no change in quality for parser output, only change in encoding
  → *same* parser output with TIGER-style encoding yields lower PARSEVAL results

- TüBa-D/Z grammar yields higher precision in the PARSEVAL metric
- TIGER yields higher recall for PARSEVAL
Experiment II: Conclusions

- Constant decrease in PARSEVAL results for modified trees is consistent with results in Kübler (2005) and Maier (2006)
- But: no change in quality for parser output, only change in encoding
  → same parser output with TIGER-style encoding yields lower PARSEVAL results
  - TüBa-D/Z grammar yields higher precision in the PARSEVAL metric
  - TIGER yields higher recall for PARSEVAL
Experiment III: Dependency-Based Evaluation

  - Convert trees into dependency relations in the form: WORD POS HEAD
  - Use functional labels in the original trees to find the head
  - Use heuristics to find the head in the parser output

- How much noise caused by the conversion method?
  - Convert gold trees to dependency relations
  - Remove functional labels from the gold trees and get dependencies
  - Evaluate dependencies for stripped gold trees against original gold trees

  TIGER: 99.65%, TüBa-D/Z: 99.13% for test sets (1024 trees)
Experiment III: Dependency-Based Evaluation

  - Convert trees into dependency relations in the form: WORD POS HEAD
  - Use functional labels in the original trees to find the head
  - Use heuristics to find the head in the parser output

- How much noise caused by the conversion method?
  - Convert gold trees to dependency relations
  - Remove functional labels from the gold trees and get dependencies
  - Evaluate dependencies for stripped gold trees against original gold trees

TIGER: 99.65%, TüBa-D/Z: 99.13% for test sets (1024 trees)
Experiment III: Dependency-Based Evaluation

- Convert trees into dependency relations in the form: WORD POS HEAD
- Use functional labels in the original trees to find the head
- Use heuristics to find the head in the parser output

How much noise caused by the conversion method?
- Convert gold trees to dependency relations
- Remove functional labels from the gold trees and get dependencies
- Evaluate dependencies for stripped gold trees against original gold trees

TIGER: 99.65%, TüBa-D/Z: 99.13% for test sets (1024 trees)
Experiment III: Dependency-Based Evaluation

  - Convert trees into dependency relations in the form:
    WORD POS HEAD
  - Use functional labels in the original trees to find the head
  - Use heuristics to find the head in the parser output

- How much noise caused by the conversion method?
  - Convert gold trees to dependency relations
  - Remove functional labels from the gold trees and get dependencies
  - Evaluate dependencies for stripped gold trees against original gold trees

TIGER: 99.65%, TüBa-D/Z: 99.13% for test sets (1024 trees)
**Experiment III: Comparing all measures**

- Same test sets (1024 sentences)
- Raw text as parser input

<table>
<thead>
<tr>
<th></th>
<th>PARSEVAL</th>
<th>LA</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>Prec</td>
</tr>
<tr>
<td>TIGER</td>
<td>81.21</td>
<td>81.04</td>
<td>93.88</td>
</tr>
<tr>
<td>TüBa</td>
<td>87.24</td>
<td>83.77</td>
<td>92.58</td>
</tr>
</tbody>
</table>

Table: Parsing results for the three evaluation metrics
Experiment III: Comparing all measures

- Same test sets (1024 sentences)
- Raw text as parser input

<table>
<thead>
<tr>
<th></th>
<th>PARSEVAL</th>
<th></th>
<th>LA</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec</td>
<td>Rec</td>
<td>Prec</td>
<td>Prec</td>
</tr>
<tr>
<td>TIGER</td>
<td>81.21</td>
<td>81.04</td>
<td>93.88</td>
<td>85.71</td>
</tr>
<tr>
<td>TüBa</td>
<td>87.24</td>
<td>83.77</td>
<td>92.58</td>
<td>76.64</td>
</tr>
</tbody>
</table>

Table: Parsing results for the three evaluation metrics
Conclusions

- Experiment I showed that trees with the same number and type of errors can yield crucially different parsing results in the PARSEVAL metric.
- Experiment II showed that when converting TüBa-D/Z trees to a TIGER-style format not the quality of the parser output but the decreasing number of nodes leads to a decrease in parsing scores.
- Experiment III showed that LA and dependency-based evaluation give better results for TIGER, while PARSEVAL favours the TüBa-DZ.
Conclusions

- Experiment I showed that trees with the same number and type of errors can yield crucially different parsing results in the PARSEVAL metric.
- Experiment II showed that when converting TüBa-D/Z trees to a TIGER-style format not the quality of the parser output but the decreasing number of nodes leads to a decrease in parsing scores.
- Experiment III showed that LA and dependency-based evaluation give better results for TIGER, while PARSEVAL favours the TüBa-DZ.
Conclusions

- Experiment I showed that trees with the same number and type of errors can yield crucially different parsing results in the PARSEVAL metric.
- Experiment II showed that when converting TüBa-D/Z trees to a TIGER-style format not the quality of the parser output but the decreasing number of nodes leads to a decrease in parsing scores.
- Experiment III showed that LA and dependency-based evaluation give better results for TIGER, while PARSEVAL favours the TüBa-DZ.
It is not clear which of the annotation schemes is more adequate to support PCFG parsing.

The question, whether German is harder to parse than English or not, is still undecided.
Conclusions (II)

- It is not clear which of the annotation schemes is more adequate to support PCFG parsing.
- The question, whether German is harder to parse than English or not, is still undecided.
Thank You!
Questions?


Drach, Erich. 1937. *Grundgedanken der Deutschen Satzlehre*. Frankfurt/M.


Telljohann, Heike, Erhard W. Hinrichs, Sandra Kübler, and Heike Zinsmeister. 2005. *Stylebook for the Tübingen Treebank of Written...*
German (TüBa-D/Z). Seminar für Sprachwissenschaft, Universität Tübingen, Germany.