Automatic Evaluation of Robustness and Degradation in Tagging and Parsing

Johnny Bigert, Ola Knutsson, Jonas Sjöbergh
Royal Institute of Technology,
Stockholm, Sweden
Contact: johnny@kth.se
Problem

NLP systems are often faced with noisy and ill-formed input:

- How do we reliably evaluate the performance of NLP systems?
- Which methods of tagging and parsing are robust?
Problem

- The performance of a NLP system is sensitive to noisy and ill-formed input
- Manual evaluations of robustness is tedious and time-consuming
- Manual evaluation is difficult to compare and reproduce
- Resources with noisy data is rare
Outline

- Introduce artificial spelling errors using software (Missplel)
- Increasing error levels will affect the NLP system performance
- Evaluation of degradation of tagging and parsing performance (AutoEval)
Introducing spelling errors

- Missplel (Bigert et al)
- Generic tool to introduce human-like spelling errors
- Highly configurable
- Language and tag set independent
- Freeware, open source
  http://www.nada.kth.se/theory/humanlang/tools.html
Introducing spelling errors

- Start with correct text (Swedish, the SUC corpus, Ejerhed et al)
- Introduce errors in, say, 10% of the words
- Spelling errors resulting in non-existing words only
- No change in parse tree
Introducing spelling errors

- 10 misspelled texts for each error level
- Eliminate the influence of chance
- Six error levels: 0%, 1%, 2%, 5%, 10%, 20%
- 15 000 words with parse info
Missplel example

Letters   NN2
ownd      VM0
be        VBI
welcome   AJ0-NN1

Litters   NN2 damerau/wordexist-notagchange
would     VM0 ok
bee       NN1 sound/wordexist-tagchange
welmoe    ERR damerau/nowordexist-tagchange
Tagging

- The texts were tagged using
  - HMM tagger (TnT, Brants)
  - Brill tagger (fnTBL, Ngai & Florian)
  - Baseline tagger (unigram)
The tagged texts were parsed using:
- GTA parser (Knutsson et al)
- Baseline parser (unigram, CoNLL)

GTA - Granska text analyzer
- Rule-based
- Hand-crafted rules
- Context-free formalism
Viktigaste (the most important) APB|NPB CLB
redskapen (tools) NPI CLI
vid (in) PPB CLI
ympning (grafting) NPB|PPI CLI
är (is) VCB CLI
annars (normally) ADV|PB CLI
papper (paper) NPB|NPB CLI
och (and) NPI CLI
penna (pen) NPB|NPI CLI
, 0 CLB
menade (meant) VCB CLI
han (he) NPB CLI
0 CLI
Evaluation

Evaluation was carried out using AutoEval (Bigert et al):

- Automated handling of plain-text and XML input/output and data storage
- Script language
- Highly configurable and extendible (C++)
- Freeware, open source
Evaluation

- Tagging:
  - **Accuracy**, correct tag if exact match

- Parsing:
  - **Accuracy**, correct row if exact match
  - **Precision** and **recall** per phrase category, correct if exact match after removing all other phrase types

- Clause boundary identification
  - **Precision** and **recall** for CLB
## Results

### Results of the tagging task (accuracy):

<table>
<thead>
<tr>
<th>Tagger</th>
<th>0%</th>
<th>1%</th>
<th>2%</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base</strong></td>
<td>85.2</td>
<td>84.4</td>
<td>83.5</td>
<td>81.2</td>
<td>77.1</td>
<td>69.0</td>
</tr>
<tr>
<td></td>
<td>(0.9)</td>
<td>(1.9)</td>
<td>(4.6)</td>
<td>(9.5)</td>
<td>(19.0)</td>
<td></td>
</tr>
<tr>
<td><strong>Brill</strong></td>
<td>94.5</td>
<td>93.8</td>
<td>93.0</td>
<td>90.9</td>
<td>87.4</td>
<td>80.1</td>
</tr>
<tr>
<td></td>
<td>(0.7)</td>
<td>(1.5)</td>
<td>(3.8)</td>
<td>(7.5)</td>
<td>(15.2)</td>
<td></td>
</tr>
<tr>
<td><strong>TnT</strong></td>
<td>95.5</td>
<td>95.0</td>
<td>94.3</td>
<td>92.4</td>
<td>89.5</td>
<td>83.3</td>
</tr>
<tr>
<td></td>
<td>(0.5)</td>
<td>(1.2)</td>
<td>(3.2)</td>
<td>(6.2)</td>
<td>(12.7)</td>
<td></td>
</tr>
</tbody>
</table>
## Results

Results of the parsing task (accuracy):

<table>
<thead>
<tr>
<th>Tagger</th>
<th>0%</th>
<th>1%</th>
<th>2%</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>81.0</td>
<td>80.2</td>
<td>79.1</td>
<td>76.5</td>
<td>72.4</td>
<td>64.5</td>
</tr>
<tr>
<td></td>
<td>(0.9)</td>
<td>(2.3)</td>
<td>(5.5)</td>
<td>(10.6)</td>
<td>(20.3)</td>
<td></td>
</tr>
<tr>
<td>Brill</td>
<td>86.2</td>
<td>85.4</td>
<td>84.5</td>
<td>82.0</td>
<td>78.0</td>
<td>70.3</td>
</tr>
<tr>
<td></td>
<td>(0.9)</td>
<td>(1.9)</td>
<td>(4.8)</td>
<td>(9.5)</td>
<td>(18.4)</td>
<td></td>
</tr>
<tr>
<td>TnT</td>
<td>88.7</td>
<td>88.0</td>
<td>87.2</td>
<td>85.2</td>
<td>81.7</td>
<td>75.1</td>
</tr>
<tr>
<td></td>
<td>(0.7)</td>
<td>(1.6)</td>
<td>(3.9)</td>
<td>(7.8)</td>
<td>(15.3)</td>
<td></td>
</tr>
</tbody>
</table>

Baseline parser: 59.2% at the 0% error level, using TnT
Conclusions

- Automated method to determine the robustness of tagging and parsing under the influence of noisy input
- No manual intervention
- Greatly simplifies repeated testing of NLP components
- Freeware
Software

- Missplel and AutoEval
- Open source
- Available for download at the Missplel and AutoEval homepage

http://www.nada.kth.se/theory/humanlang/tools.html