Language Technology for Polish in Practice
Morpho-syntactic processing of Polish

Maciej Piasecki
Wrocław University of Science and Technology
G4.19 Research Group
maciej.piasecki@pwr.wroc.pl
2017-01-17
Tools

- **Morphological analysis:**
  - Morfeusz – IPI PAN (Woliński, 2014)
- **Morpho-syntactic analysis**
  - TaKIPI – PWr., old but still used (Piasecki, 2007)
  - WCRFT 1 and WCRFT 2 – PWr. (Radziszewski, 2013)
    - Available via CLARIN-PL Web Services
  - Pantera – IPI PAN (Acedański, 2010)
  - Concraft – IPI PAN (Waszczuk, 2012)
- **Shallow parsing and chunking**
  - Iobber – PWr., trained on KPWr (Radziszewski & Pawlaczek, 2013)
  - Spejd – IPI PAN (Przepiórkowski, 2008)
TaKIPI – the first tagger for Polish

- TaKIPI (Piasecki & Godlewska, 2006)
  - name comes from: ‘a tagger for the IPI PAN Corpus’
  - the first morpho-syntactic tagger for Polish publicly available and in wider use
  - quite fast and easy to install, so still in use
  - partial disambiguation (in some cases more than one tag per word is left), combined with simple morphological guesser
  - ~93% of weak accuracy

- Construction
  - tiered tagging: gradual disambiguation during several phases
  - heterogenous: limited set of rules combined with decision trees
TaKIPI: tiered tagging

- Text
- Morfeusz
- Reader & Pre-sentencer
- Filter of Rules
- Unigram Classifier
- Manager of Classifiers
- Annotator
- Sentencer & Writer
- Tagged sentences

- Hand-written rules
- Unigram dictionary

- Dictionary of abbreviations
- Layer masks
- Decision trees for layers

- DT₁
- DDₙ
TaKIPi: tiered tagging

Unigram Classifier

Manager of Classifiers

Package Cut-off

Normalisation

layer masks

decision trees for layers

DTs layer$_1$

... DTs layer$_n$

next layer?
WCRFT tagger

- WCRFT comes from `Wrocław CRF-based Tagger’ (Radziszewski 2013)
  - morpho-syntactic tagger for Polish
  - all words, tags for unknown guessed, but not their lemmas
- Construction
  - tiered tagging in several phases
  - based on supervised learning and NKJP (Polish National Corpus)
    - CRF algorithm (Conditional Random Fields)
    - 1 mln word part of NKJP used for training
WCRFT: feature types

1. word form of a token,
2. possible values of the grammatical class of a token,
3. possible values of grammatical number,
4. possible values of gender,
5. possible values of grammatical case,
6. a predicate checking if there holds a grammatical agreement of the current and the next token with respect to number, gender and case,
7. a similar predicate that checks the agreement of the previous, current and the next tokens (-1, 0, 1),
8. if the current token’s orthographic form starts with an upper-case letter,
9. if it starts with lower-case letter.
1. Word forms for $p(ositions) \in \{-2, -1, 0, 1, 2\}$
2. Word form bigrams: $(-1, 0)$ and $(0, 1)$
3. Grammatical class for $p \in \{-2, -1, 0, 1, 2\}$
4. Class bigrams: $(-2, -1), (-1, 0), (0, 1), (1, 2)$
5. Class trigrams: $(p - 1, p, p + 1)$ for $p \in \{-2, 0, 1\}$
6. Case: for $p \in \{-2, -1, 0, 1, 2\}$
7. Gender: for $p \in \{-2, -1, 0, 1, 2\}$
8. Number: for $p \in \{-2, -1, 0, 1, 2\}$
9. Agreement: $\text{num\_gen\_cas}(-1, 0), \text{num\_gen\_cas}(-1 \ldots 1)$
10. Agreement: $\text{num\_gen\_cas}(-1 \ldots 1)$
11. Upper case letter: $p=0$
WCRFT: implementation in brief

- Problems with NKJP (Polish National Corpus) as training source
  - … permanent, to be honest
  - some inconsistencies in the morphological annotation
  - Increasing differences between Morfeusz and NKJP
    - output format – from the very beginning
    - tagset, evolving, e.g. new grammatical classes in Morfeusz
    - sets of tags for word forms – increasing lack of synchronisation
- Attempt to decrease their negative influence in WCRFT
  - if the tag assigned manually in NKJP is generated by Morfeusz, it is used
  - otherwise the word is marked as unknown (ign) and assigned the tag from NKJP
Training
- Tag sets for unknown words are collected from NKJP
- performed in tiers (phase): grammatical class, number, case, … (other categories)
  - one attribute per tier (phase)
  - partial disambiguation after each tier: all tags inconsistent with the choice eliminated - ideal decision from NKJP
- feature templates are used to generate characteristic functions for CRF++
  - WCCL language is used to define feature templates
  - Corpus2 library is used for reading corpus/text

Tagging
- in tiers, but partial disambiguation on the basis of the tagger results
### WCRFT evaluation

<table>
<thead>
<tr>
<th>Tagger</th>
<th>$Acc_{lower}$</th>
<th>$Acc_{upper}$</th>
<th>$Acc_{lower}^K$</th>
<th>$Acc_{lower}^U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMBT</td>
<td>87.50%</td>
<td>87.82%</td>
<td>89.78%</td>
<td>13.57%</td>
</tr>
<tr>
<td>PANTERA</td>
<td>88.79%</td>
<td>89.09%</td>
<td>91.08%</td>
<td>14.70%</td>
</tr>
<tr>
<td>WMBT+u</td>
<td>89.71%</td>
<td>90.04%</td>
<td>91.20%</td>
<td>41.45%</td>
</tr>
<tr>
<td>WCRFT</td>
<td>90.34%</td>
<td>90.67%</td>
<td>91.89%</td>
<td>40.13%</td>
</tr>
</tbody>
</table>

- Evaluated on the manually annotated part of NKJP (average from 10 fold cross-validation)


Adam Radziszewski, Adam Pawlaczek, (2013) „Incorporating head recognition into a CRF chunker”. In: IIS 2013, Warsaw, Poland, June 17-18, 2013. [link]

Resource: [link]

Thank you very much for your attention!

www.clarin-pl.eu