Learning Morphology from the Corpus

Ondřej Dušek

Institute of Formal and Applied Linguistics
Charles University in Prague

November 11, 2013
Motivation (general)

Morphology needed in most NLP tasks

- Parsing
- Structural MT
- Factored phrase-based MT
- Corpora
- User interfaces
- Dialogue systems

Morphology module influences overall quality of the systems
Motivation (personal)

“Avoid the X@ tag in Czech as much as possible”

- Words unknown to the Czech dictionary are relatively common in some applications
Motivation (personal)

“Avoid the X@ tag in Czech as much as possible”

- Words unknown to the Czech dictionary are relatively common in some applications
  - KHRESMOI – translation of medical text: terms
  - ALEX dialogue system – public transport: stop names
- Up to 5% of words are not recognized in special domains

Dolnokrčská X@------------
arthroplastika X@-------------
Motivation (personal)

“Avoid the X@ tag in Czech as much as possible”

- Words unknown to the Czech dictionary are relatively common in some applications
  - KHRESMOI – translation of medical text: terms
  - ALEX dialogue system – public transport: stop names
- Up to 5% of words are not recognized in special domains
- There's no guesser in Treex (that I know of)

Dolnokrčská X@---------------
artroplastika X@---------------
Motivation (personal)

“Avoid the X@ tag in Czech as much as possible”

- Words unknown to the Czech dictionary are relatively common in some applications
  - KHRESMOI – translation of medical text: terms
  - ALEX dialogue system – public transport: stop names
- Up to 5% of words are not recognized in special domains
- There's no guesser in Treex (that I know of)

“Inflect anything”

- Translate and create unseen phrases
- Speak freely in dialogue systems
Exploiting the regularities in morphology

- Morphology of many languages is mostly regular, but for a certain number of exceptions
- Size, number, and shape of inflection patterns differ
Exploiting the regularities in morphology

- Morphology of many languages is mostly regular, but for a certain number of exceptions
- Size, number, and shape of inflection patterns differ

Proportion in Grammar

<table>
<thead>
<tr>
<th>Past Tense</th>
<th>Past Participle</th>
</tr>
</thead>
<tbody>
<tr>
<td>grew</td>
<td>grown</td>
</tr>
<tr>
<td>flew</td>
<td>?</td>
</tr>
</tbody>
</table>

\[
grew \cdot \frac{flew}{grown} = \frac{flown}{grew}
\]
Possible approaches to morphology

Dictionaries?

- Work well, reliable
- Limited coverage and/or availability
Possible approaches to morphology

Dictionaries?

- Work well, reliable
- Limited coverage and/or availability

Hand-written rules?

- Hard to maintain with complex morphology
Possible approaches to morphology

**Dictionaries?**
- Work well, reliable
- Limited coverage and/or availability

**Hand-written rules?**
- Hard to maintain with complex morphology

**Learning from the data!**
- Obtaining the rules automatically
- Plenty of corpora of sufficient size available
My experiments with morphology

- in chronological (less logical) order
My experiments with morphology

- in chronological (less logical) order

1. Generation

- with Filip Jurčíček (see also: our paper at ACL–SRW 2013)
- *Flect*: statistical morphology generator
My experiments with morphology

- in chronological (less logical) order

1. Generation

- with Filip Jurčíček (see also: our paper at ACL–SRW 2013)
- *Flect*: statistical morphology generator

2. Analysis

- recent, only partially finished experiments on Czech
- a simple morphology module to go with the *Featurama* tagger, comparison with others
My experiments with morphology

- in chronological (less logical) order

1. Generation
   - with Filip Jurčíček (see also: our paper at ACL-SRW 2013)
   - *Flect*: statistical morphology generator

2. Analysis
   - recent, only partially finished experiments on Czech
   - a simple morphology module to go with the *Featurama* tagger, comparison with others

3. Discussion
Flect: Morphology generator

- Using machine learning to predict inflection
**Flect: Morphology generator**

- Using machine learning to predict inflection
- Only previous statistical morphology module known to us: *Bohnet et al. (2010)*
**Flect: Morphology generator**

- Using machine learning to predict inflection
- Only previous statistical morphology module known to us: *Bohnet et al. (2010)*
- *Flect* tested on 6 languages from the CoNLL 2009 data set with a varying degree of morphological richness
The need to generate morphology

- English – not so much:
  hard-coded solutions often work well enough
The need to generate morphology

- English – not so much: hard-coded solutions often work well enough
- Languages with more inflection (e.g. Czech): even the simplest applications have trouble with morphology

Toto se líbí uživateli Jana Nováková.

This is liked by user [masc] (name) [fem] 

Děkujeme, Jan Novák, vaše hlasování bylo vytvořeno.

Thank you, (name) [nom] your poll has been created
The task at hand

- Input: Lemma (base form) or stem + morphological properties (POS, case, gender, etc.)
- Output: Inflected word form
- Inverse to POS tagging

\[
\begin{align*}
\text{word} & \quad + \quad \text{NNS} \quad \rightarrow \quad \text{words} \\
\text{Wort} & \quad + \quad \text{NN} \quad \text{Neut, Pl, Dat} \quad \rightarrow \quad \text{Wörtern} \\
\text{be} & \quad + \quad \text{VBZ} \quad \rightarrow \quad \text{is} \\
\text{ser} & \quad + \quad \text{V} \quad \text{gen=c, num=s, person=3, mood=indicative, tense=present} \quad \rightarrow \quad \text{es}
\end{align*}
\]
Casting inflection patterns as multi-class classification

Our inflection rules: *edit scripts*

- **A kind of diffs**: how to modify the lemma to get the form
- Based on Levenshtein distance
Casting inflection patterns as multi-class classification

Our inflection rules: *edit scripts*

- **A kind of diffs**: how to modify the lemma to get the form
- Based on Levenshtein distance
Casting inflection patterns as multi-class classification

Our inflection rules: *edit scripts*

- **A kind of diffs:** how to modify the lemma to get the form
- Based on Levenshtein distance
Casting inflection patterns as multi-class classification

Our inflection rules: edit scripts

- **A kind of diffs:** how to modify the lemma to get the form
- Based on Levenshtein distance
Features useful for morphology generation

- Same POS + same ending = (often) same inflection

```yaml
sky + NNS ➞ -ies
fly
bind + VBD ➞ -ound
find
```
Features useful for morphology generation

- Same POS + same ending = (often) same inflection

- Suffixes = good features to generalize to unseen inputs
- Machine learning should be able to deal with counter-examples
Features useful for morphology generation

- Same POS + same ending = (often) same inflection
  
  \[
  \text{sky} + \text{sky} \rightarrow \text{flies}
  \]
  
  \[
  \text{fly} + \text{NNS} \rightarrow \text{-ies}
  \]
  
  \[
  \text{bind} + \text{VBD} \rightarrow \text{-ound}
  \]

- Suffixes = good features to generalize to unseen inputs
- Machine learning should be able to deal with counter-examples
- Capitalization: no influence on morphology
Our system *Flect*: Overall procedure
Our system *Flect*: Overall procedure

1. Get **features** from lemma, POS, suffixes
   (+morph. properties & their combinations, possibly context)
Our system *Flect*: Overall procedure

1. Get **features** from lemma, POS, suffixes (+morph. properties & their combinations, possibly context)
2. Predict **edit scripts** using Logistic regression

![Diagram showing the overall procedure of the Flect system]
Our system *Flect*: Overall procedure

1. Get **features** from lemma, POS, suffixes (+morph. properties & their combinations, possibly context)
2. Predict **edit scripts** using Logistic regression
3. Use them as rules to obtain **form** from lemma
Testing *Flect* on 6 languages

- **CoNLL 2009 data:** varying morphology richness & tagsets
Testing *Flect* on 6 languages

- **CoNLL 2009 data**: varying morphology richness & tagsets

<table>
<thead>
<tr>
<th>Language</th>
<th>Total Accuracy</th>
<th>Unseen Forms Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>92</td>
<td>90</td>
</tr>
<tr>
<td>German</td>
<td>94</td>
<td>100</td>
</tr>
<tr>
<td>Czech</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>Spanish</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>Japanese</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>German</td>
<td>92</td>
<td></td>
</tr>
</tbody>
</table>

Works well even on unseen forms: suffixes help overcome generalization errors, e.g. *torpedo* + *VBN* = *torpedone*.
Testing *Flect* on 6 languages

- **CoNLL 2009 data**: varying morphology richness & tagsets

![Accuracy (%)](image)

- Works well even on unseen forms: suffixes help
Testing *Flect* on 6 languages

- **CoNLL 2009 data**: varying morphology richness & tagsets

- Works well even on unseen forms: suffixes help
  - over-generalization errors, e.g. *torpedo* + *VBN* = *torpedone*
  - German: syntax-sensitive morphology
**Flect vs. a dictionary from the same data**

- English: Dictionary gets OK relatively soon
**Flect vs. a dictionary from the same data**

- English: Dictionary gets OK relatively soon
- Czech: Dictionary fails on unknown forms, our system works

![Graph showing accuracy and error reduction](#)
Flect vs. a dictionary from the same data

- English: Dictionary gets OK relatively soon
- Czech: Dictionary fails on unknown forms, our system works

<table>
<thead>
<tr>
<th>Training data part (%)</th>
<th>Dictionary (Total)</th>
<th>Dictionary (Unknown forms)</th>
<th>Flect (Total)</th>
<th>Flect (Unknown forms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>92.88</td>
<td>98.25</td>
<td>99.45</td>
<td>99.45</td>
</tr>
</tbody>
</table>

92% error reduction
Conclusions (morphology generation)

General observations:

- Inflection rules/patterns can be learned from a corpus
- Suffix features are useful to inflect unseen words
- Detailed morphological features and context features help
Conclusions (morphology generation)

General observations:

- Inflection rules/patterns can be learned from a corpus
- Suffix features are useful to inflect unseen words
- Detailed morphological features and context features help

Our system *Flect*:

- improves on a dictionary learnt from the same data
- gains more in morphologically rich languages (Czech)
- can be combined with a dictionary as a back-off for OOVs
The task of finding the right lemma (stem/base form) and part-of-speech tag for a word form can be (and is) divided into:

1. **Morphological analysis**
   - Finding all possible POS tags / lemmas for the word form

2. **Tagging**
   - Selecting the one correct POS tag / lemma for the word form according to the context

Lemmas are sometimes predicted separately from POS tags (or not at all); we try to predict lemmas and tags together.
Morphological analysis/Tagging

The task of finding the right lemma (stem/base form) and part-of-speech tag for a word form can be (and is) divided into:

1. Morphological analysis
   finding all possible POS tags / lemmas for the word form

\[ \text{ženu} \quad \text{žena} \quad \text{NNFS4}---\text{A}----- \]
\[ \text{hnát} \quad \text{VB-S}---\text{1P-AA}--- \]
Morphological analysis/Tagging

The task of finding the right lemma (stem/base form) and part-of-speech tag for a word form can be (and is) divided into:

1. **Morphological analysis**
   finding **all possible** POS tags / lemmas for the word form

2. **Tagging**
   selecting the one correct POS tag / lemma for the word form according to the context

<table>
<thead>
<tr>
<th>Word</th>
<th>Lemma</th>
<th>POS Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>ženu</td>
<td>žena</td>
<td>NNFS4----A----</td>
</tr>
<tr>
<td>hnat</td>
<td>VB-S---1P-AA---</td>
<td>✖</td>
</tr>
</tbody>
</table>
Morphological analysis/Tagging

The task of finding the right lemma (stem/base form) and part-of-speech tag for a word form can be (and is) divided into:

1. **Morphological analysis**
   finding **all possible** POS tags / lemmas for the word form

2. **Tagging**
   selecting the one correct POS tag / lemma for the word form according to the context

Lemmas are sometimes predicted separately from POS tags (or not at all); we try to predict lemmas and tags together.

ženu  žena  NNFS4------A-----  ✓
hnát  VB-S---1P-AA---  ✗
A side note

Lemma simplifications compared to Hajič (2004)'s morphological dictionary:

Tatra-2_;R_^(vozidlo)
A side note

Lemma simplifications compared to Hajič (2004)'s morphological dictionary:

1. No lemma “tails” (AddInfo)

Tatra–2_;R_^(vozidlo)
A side note

Lemma simplifications compared to Hajič (2004)'s morphological dictionary:

1. No lemma “tails” (AddInfo)
2. Lemmas are case-insensitive

**tatra-2_\text{;}R_\wedge(\text{vozidlo})**
A side note

Lemma simplifications compared to Hajič (2004)'s morphological dictionary:

1. No lemma “tails” (AddInfo)
2. Lemmas are case-insensitive

This enables us to learn the lemmas from data (while generating from such lemmas is still possible).

tatra\textsuperscript{2\_;}R\textsuperscript{\_\_\_\_}(vozidlo)
Learning morphological analysis from the data

- Parallel to learning generation
  - We can use similar edit scripts (reversed: form to lemma)

nejhezčímu  >4-ký, <nej
hezký

[replace ending]  [remove beginning]
Learning morphological analysis from the data

- Parallel to learning generation
  - We can use similar edit scripts (reversed: form to lemma)
    - \texttt{nejhezčímu} \rightarrow >4-ký, <nej
    - \texttt{hezký} [replace ending]
    - [remove beginning]

- Not so new – some of the previous systems:
  - \textit{Hajič (2004)}: statistical guesser (for forms that are not in the dictionary)
  - \textit{Chrupała et al. (2008)} – \textit{Morfette}: completely statistical (predicting probability distributions for lemmas and tags + global optimization)
My experiments

Preconsiderations

- only analysis (leave the hard work to the tagger)
- for all words (no dictionary needed)
My experiments

Preconsiderations

- only analysis (leave the hard work to the tagger)
- for all words (no dictionary needed)

The Solution

- Just memorize suffixes of certain length with tags + lemma edit-scripts
  - No machine learning here (pass all variants matching the suffix to the tagger)
  - Similar to Hajič (2004)'s guesser

... "ebí": {"|NNNS1-----A----",
          "|NNNS6-----A----",
          ">1-it|VB-S---3P-AA---",
          ">1-it|VB-P---3P-AA---",
          "|Db-------------" }, ...
My experiments

Preconsiderations

- only analysis (leave the hard work to the tagger)
- for all words (no dictionary needed)

The Solution

- Just memorize suffixes of certain length with tags + lemma edit-scripts
  - No machine learning here
    (pass all variants matching the suffix to the tagger)
  - Similar to Hajič (2004)’s guesser

- Small improvements: smoothing, irregular words remembered as a whole
- Parameters: length of suffixes, occurrence count threshold

"ebí": {
  "NNNS1-----A----",
  "NNNS6-----A----",
  ">1-it|VB-S---3P-AA---",
  ">1-it|VB-P---3P-AA---",
  "Db-------------"}
...
Results: Morphological analysis

Coverage (recall) measured on the PDT 2.5 development test set (lemmas lowercased, no AddInfo)

<table>
<thead>
<tr>
<th>Method</th>
<th>cov (%)</th>
<th>φ sugg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hajič (060406)</td>
<td>98.82</td>
<td>3.85</td>
</tr>
<tr>
<td>Hajič (060406) + guesser</td>
<td>99.35</td>
<td>4.06</td>
</tr>
<tr>
<td>Hajič (131023)</td>
<td>98.52</td>
<td>4.00</td>
</tr>
<tr>
<td>Hajič (131023) + guesser</td>
<td>99.01</td>
<td>4.18</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4)</td>
<td>98.71</td>
<td>5.69</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3)</td>
<td>99.30</td>
<td>11.83</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4, thr 2)</td>
<td>98.07</td>
<td>4.75</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3, thr 2)</td>
<td>98.91</td>
<td>9.27</td>
</tr>
</tbody>
</table>

Coverage quite OK, but a lot of false positives.
Results: Morphological analysis

Coverage (recall) measured on the PDT 2.5 development test set (lemmas lowercased, no AddInfo)

<table>
<thead>
<tr>
<th></th>
<th>cov (%)</th>
<th>φ sugg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hajič (060406)</td>
<td>98.82</td>
<td>3.85</td>
</tr>
<tr>
<td>Hajič (060406) + guesser</td>
<td>99.35</td>
<td>4.06</td>
</tr>
<tr>
<td>Hajič (131023)</td>
<td>98.52</td>
<td>4.00</td>
</tr>
<tr>
<td>Hajič (131023) + guesser</td>
<td>99.01</td>
<td>4.18</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4)</td>
<td>98.71</td>
<td>5.69</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3)</td>
<td>99.30</td>
<td>11.83</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4, thr 2)</td>
<td>98.07</td>
<td>4.75</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3, thr 2)</td>
<td>98.91</td>
<td>9.27</td>
</tr>
</tbody>
</table>

Coverage quite OK, but a lot of false positives.
## Results: Tagging

Taggers trained on PDT 2.5 (training + development set), tested on the evaluation set (accuracy in %).

<table>
<thead>
<tr>
<th>analysis</th>
<th>tagger</th>
<th>tag</th>
<th>lemma</th>
<th>joint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hajič (060406)</td>
<td>Featurama</td>
<td>95.38</td>
<td>99.27</td>
<td>95.29</td>
</tr>
<tr>
<td>Hajič (060406) + guesser</td>
<td></td>
<td>95.77</td>
<td>99.31</td>
<td>95.64</td>
</tr>
<tr>
<td>Hajič (131023)</td>
<td></td>
<td>95.15</td>
<td>99.13</td>
<td>94.95</td>
</tr>
<tr>
<td>Hajič (131023) + guesser</td>
<td></td>
<td>95.49</td>
<td>99.18</td>
<td>95.26</td>
</tr>
<tr>
<td>Milan Straka's tagger beta (131023)</td>
<td></td>
<td>94.72</td>
<td>99.13</td>
<td>94.53</td>
</tr>
<tr>
<td>Milan Straka's tagger beta (131023) + guesser</td>
<td></td>
<td>95.07</td>
<td>99.15</td>
<td>94.85</td>
</tr>
<tr>
<td>Morfette (trained on tamw only)</td>
<td></td>
<td>89.79</td>
<td>97.65</td>
<td>89.39</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4)</td>
<td>Featurama</td>
<td>94.12</td>
<td>97.80</td>
<td>93.34</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3)</td>
<td></td>
<td>94.28</td>
<td>96.84</td>
<td>92.59</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4, thr 2)</td>
<td></td>
<td>93.64</td>
<td>97.86</td>
<td>93.09</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3, thr 2)</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Results: Tagging

Taggers trained on PDT 2.5 (training + development set), tested on the evaluation set (accuracy in %).

<table>
<thead>
<tr>
<th>analysis</th>
<th>tagger</th>
<th>tag</th>
<th>lemma</th>
<th>joint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hajič (060406)</td>
<td>Featurama</td>
<td>95.38</td>
<td>99.27</td>
<td>95.29</td>
</tr>
<tr>
<td>Hajič (060406) + guesser</td>
<td></td>
<td>95.77</td>
<td>99.31</td>
<td>95.64</td>
</tr>
<tr>
<td>Hajič (131023)</td>
<td></td>
<td>95.15</td>
<td>99.13</td>
<td>94.95</td>
</tr>
<tr>
<td>Hajič (131023) + guesser</td>
<td></td>
<td>95.49</td>
<td>99.18</td>
<td>95.26</td>
</tr>
<tr>
<td>Milan Straka's tagger beta (131023)</td>
<td></td>
<td>94.72</td>
<td>99.13</td>
<td>94.53</td>
</tr>
<tr>
<td>Milan Straka's tagger beta (131023) + guesser</td>
<td></td>
<td>95.07</td>
<td>99.15</td>
<td>94.85</td>
</tr>
<tr>
<td>Morfette (trained on tamw only)</td>
<td></td>
<td>89.79</td>
<td>97.65</td>
<td>89.39</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4)</td>
<td>Featurama</td>
<td>94.12</td>
<td>97.80</td>
<td>93.34</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3)</td>
<td></td>
<td>94.28</td>
<td>96.84</td>
<td>92.59</td>
</tr>
<tr>
<td>Memo-Suffixes (len 4, thr 2)</td>
<td></td>
<td>93.64</td>
<td>97.86</td>
<td>93.09</td>
</tr>
<tr>
<td>Memo-Suffixes (len 3, thr 2)</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Prof. Hajič's analysis with guesser is the best option.
Thank you for your attention

Comments and suggestions are welcome

Referenced works
Bohnet, B. et al. (2010). Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. *COLING*

Chrupała, G. et al. (2008). Learning morphology with Morfette. *LREC*


The *Flect* generator is available for download:

Contact me:
odusek@ufal.mff.cuni.cz, office 424