Robust Multilingual Statistical Morphology Generation Models

Ondřej Dušek and Filip Jurčíček

Institute of Formal and Applied Linguistics
Charles University in Prague

August 6, 2013
Introduction

Morphology in NLG

• Last step of the whole NLG pipeline
• Usually does not get a lot of attention, but is necessary
Introduction

Morphology in NLG

• Last step of the whole NLG pipeline
• Usually does not get a lot of attention, but is necessary

What we do (Flect)
Introduction

Morphology in NLG

- Last step of the whole NLG pipeline
- Usually does not get a lot of attention, but is necessary

What we do (Flect)

Semantics

Syntax

Morphology

Text

We solve this

In these languages

EN DE ES CA JA CS

Natural Language Generation

We solve this
The need for morphology in generation

• English – not so much: 
hard-coded solutions often work well enough
The need for morphology in generation

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

Toto se líbí uživateli Jana Nováková.  
This is liked by user [masc] [dat] (name) [fem] [nom]

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
Possible solutions

Dictionary?

- Works well, but has limited size
- Not many large-coverage openly available ones
Possible solutions

Dictionary?

- Works well, but has limited size
- Not many large-coverage openly available ones

Hand-written rules?

- Work well, but are hard to maintain
Possible solutions

Dictionary?
• Works well, but has limited size
• Not many large-coverage openly available ones

Hand-written rules?
• Work well, but are hard to maintain

Machine learning!
• Obtain the rules automatically
• Plenty of treebanks of sufficient size available
• Only work known to us: Bohnet et al. 2010
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

- Suffixes = good features to generalize to unseen inputs

- Machine learning should be able to deal with counter-examples

- Capitalization: no influence on morphology
Features useful for morphology generation

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

sky fly $\rightarrow$ -ies
bind find $\rightarrow$ -ound

- 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

  \[
  \text{sky} + \text{NNS} \rightarrow \text{-ies} \\
  \text{fly} + \text{VBD} \rightarrow \text{-ound}
  \]

• 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

1. Get features from lemma, POS, suffixes (including morphological properties and their combinations, possibly context)

2. Predict edit scripts using Logistic regression

3. Use them as rules to obtain form from lemma
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
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

![Bar chart showing accuracy for CoNLL 2009 data across 6 languages (EN, CS, JA, CA, ES, DE). The chart displays accuracy in percentage for both total and unseen forms.](chart.png)
Testing *Flect* on 6 languages

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

![Accuracy Chart]

- **Works well even on unseen forms**: suffixes help

---

Ondřej Dušek and Filip Jurčíček

Robust Multilingual Statistical Morphology Generation Models
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>98</td>
</tr>
<tr>
<td>Spanish</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Japanese</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Italian</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>French</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dutch</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Training Data Part (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>English</td>
</tr>
<tr>
<td></td>
<td>Dictionary</td>
</tr>
<tr>
<td></td>
<td>Flect</td>
</tr>
<tr>
<td>58%</td>
<td></td>
</tr>
<tr>
<td>76% error reduction</td>
<td></td>
</tr>
<tr>
<td>58% error reduction</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training Data Part (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Czech</td>
</tr>
<tr>
<td></td>
<td>Dictionary</td>
</tr>
<tr>
<td></td>
<td>Flect</td>
</tr>
<tr>
<td>92%</td>
<td></td>
</tr>
<tr>
<td>40% error reduction</td>
<td></td>
</tr>
<tr>
<td>58% error reduction</td>
<td></td>
</tr>
</tbody>
</table>
Flect vs. a dictionary from the same data

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

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

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
Thank you for your attention

You may download Flect (and these slides) at:
http://ufal.mff.cuni.cz/~odusek/flect/

The system is based on Python and Scikit-Learn.

You may contact us:
Ondřej Dušek & Filip Jurčíček
Charles University in Prague
odusek@ufal.mff.cuni.cz