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*)

Semantics → Syntax → Morphology → Text

We solve this

In these languages

Ondřej Dušek and Filip Jurčíček

Robust Multilingual Statistical Morphology Generation Models
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

In these languages

We solve this

EN DE ES CA JA CS
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] (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

sky + NNS → -ies
fly + NNS → -ies
bind + VBD → -ound
find + VBD → -ound

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

\[
\begin{align*}
\text{sky} + \text{NNS} & \rightarrow -\text{ies} \\
\text{fly} + \text{VBD} & \rightarrow -\text{ound}
\end{align*}
\]

- 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

- 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 (and morph. properties & 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

![Diagram showing the process of transforming a word with edit scripts](image-url)
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

Works well even on unseen forms: suffixes help over-generalization errors, e.g. *torpedo* → *torpedone*

German: syntax-sensitive morphology

Ondřej Dušek and Filip Jurčíček

Robust Multilingual Statistical Morphology Generation Models
Testing *Flect* on 6 languages

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

![Bar chart showing accuracy (%) for various languages.](chart.png)

- English: 92%
- German: 94%
- Czech: 96%
- Spanish: 98%
- Japanese: 90%
- Total: 100%

Works well even on unseen forms: suffixes help over-generalization errors, e.g. "torpedo + VBN = torpedone".

German: syntax-sensitive morphology

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>English</th>
<th>German</th>
<th>Czech</th>
<th>Spanish</th>
<th>French</th>
<th>Italian</th>
<th>Japanese</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy (%)</td>
<td>92</td>
<td>94</td>
<td>96</td>
<td>98</td>
<td>90</td>
<td>100</td>
<td>90</td>
<td></td>
</tr>
</tbody>
</table>

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

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

![Accuracy chart](chart.png)

- 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

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

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

Introduction
The system
Results

accuracy (%)

100
90
80
70
60
50
40

training data part (%)

0,1 0,5 1 5 10 20 30 50 75 100

Dictionary (Total)
Dictionary (Unknown forms)
Flect (Total)
Flect (Unknown forms)

92% error reduction
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