

Robust Multilingual Statistical Morphology Generation Models

Ondřej Dušek and Filip Jurčiček

Institute of Formal and Applied Linguistics
Charles University in Prague

August 6, 2013

Introduction

Morphology in NLG

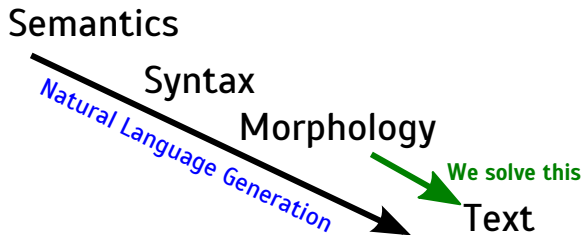
- Last step of the whole NLG pipeline
- Usually does not get a lot of attention, but is necessary

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*)

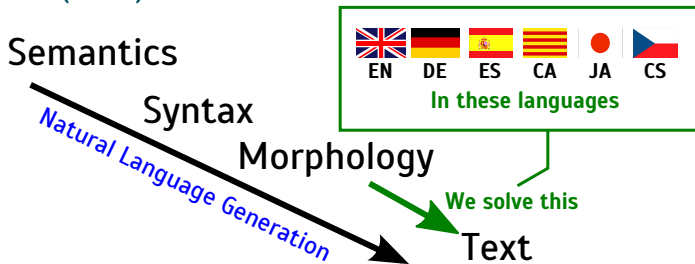


Introduction

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*)





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]
[dat] [nom]

 Děkujeme, Jan Novák^e, vaše hlasování^u
Thank you, (name)[nom] bylo vytvořeno.
your poll has been created

The task at hand

word + NNS → words
Wort + NN Neut,Pl,Dat → Wörtern

be + VBZ → is
ser + V<sup>gen=c,num=s,person=3,
mood=indicative,tense=present</sup> → es

- 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

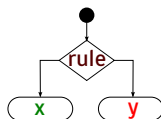
Dictionary?

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



Hand-written rules?

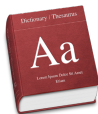
- Work well, but are hard to maintain



Possible solutions

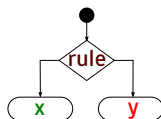
Dictionary?

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



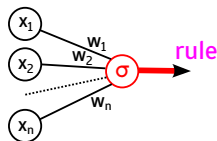
Hand-written rules?

- Work well, but are hard to maintain

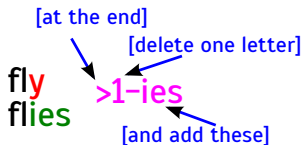


Machine learning!

- Obtain the rules automatically
- Plenty of treebanks of sufficient size available
- Only work known to us: *Bohnet et al. 2010*



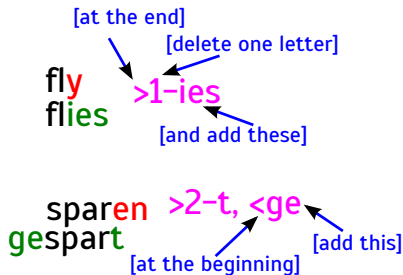
Casting inflection patterns as multi-class classification



Our inflection rules: *edit scripts*

- **A kind of diffs:** how to modify the lemma to get the form
- Based on Levenshtein distance

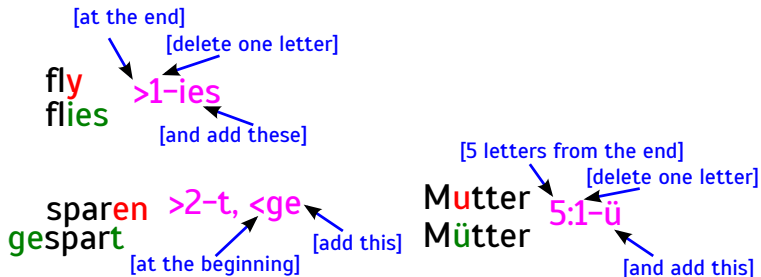
Casting inflection patterns as multi-class classification



Our inflection rules: *edit scripts*

- **A kind of diffs:** how to modify the lemma to get the form
- Based on Levenshtein distance

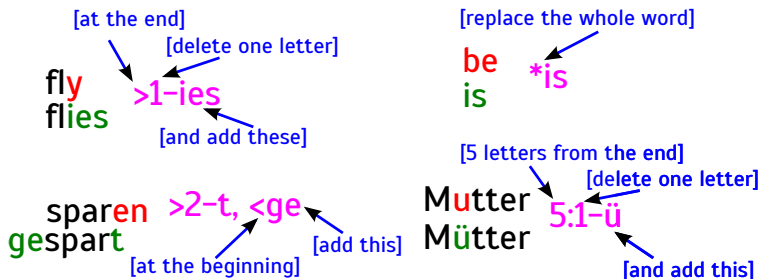
Casting inflection patterns as multi-class classification



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

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

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**

Our system *Flect*: Overall procedure

Wort

NN

PI

Neut

Dat

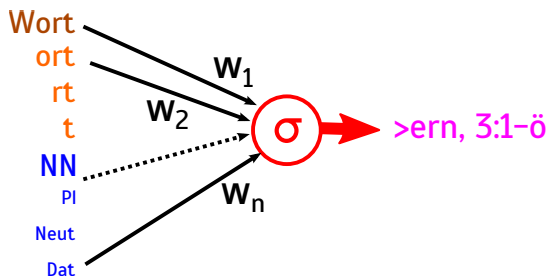
Our system *Flect*: Overall procedure

1. Get **features** from lemma, POS, suffixes
(+morph. properties & their combinations, possibly context)

Wort
ort
rt
t
NN
PI
Neut
Dat

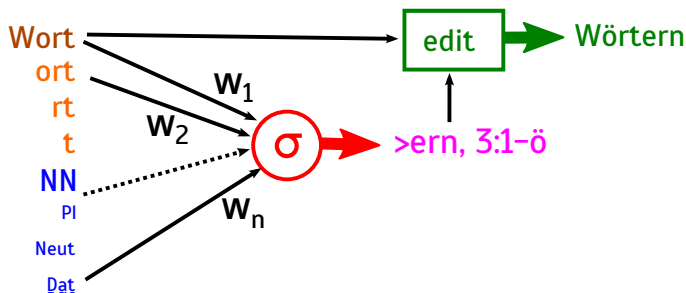
Our system *Flect*: Overall procedure

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

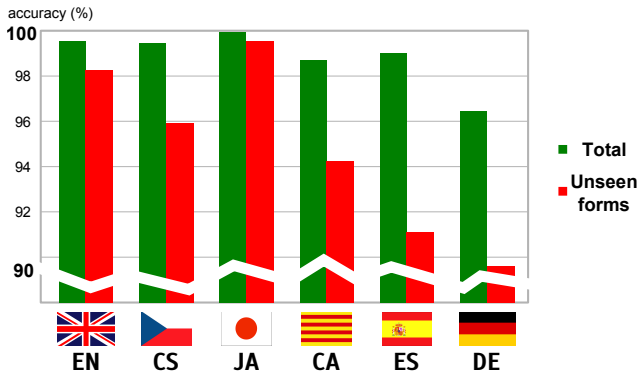


Testing *Flect* on 6 languages

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

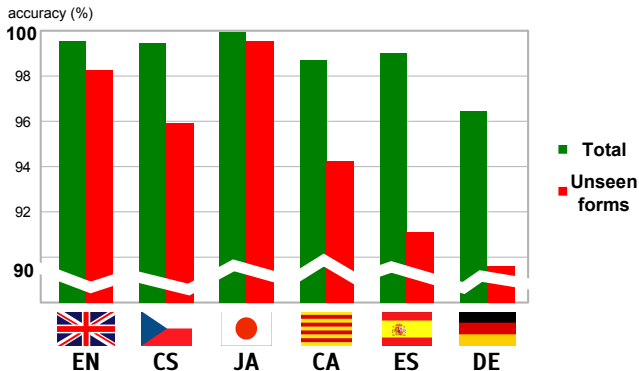
Testing *Flect* on 6 languages

- CoNLL 2009 data: varying morphology richness & tagsets



Testing *Flect* on 6 languages

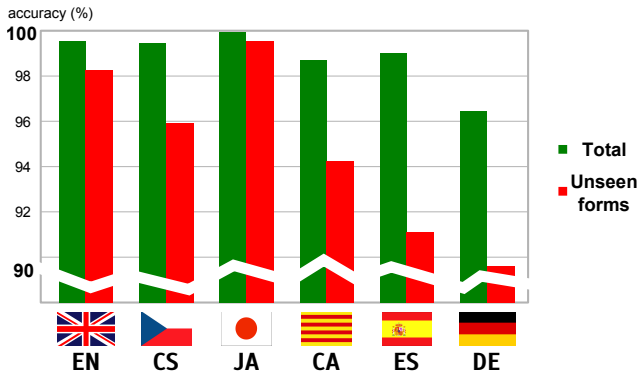
- CoNLL 2009 data: varying morphology richness & tagsets



- Works well even on unseen forms: suffixes help

Testing *Flect* on 6 languages

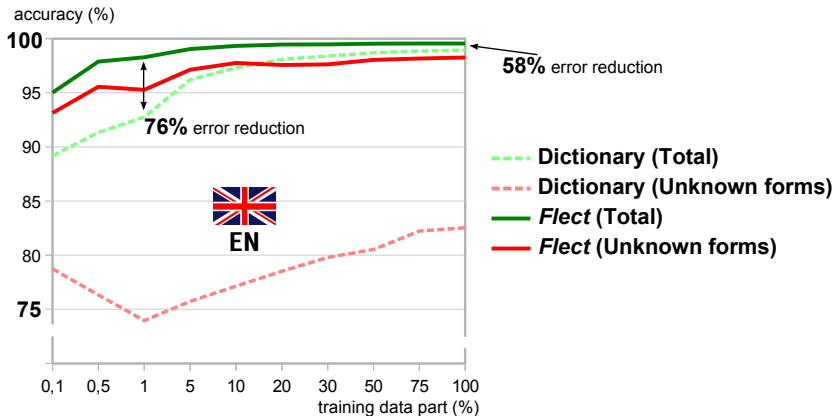
- CoNLL 2009 data: varying morphology richness & tagsets



- Works well even on unseen forms: suffixes help
 - over-generalization errors, e.g. **torpedo** + **VBN** = **torpedone**
 - German: syntax-sensitive morphology

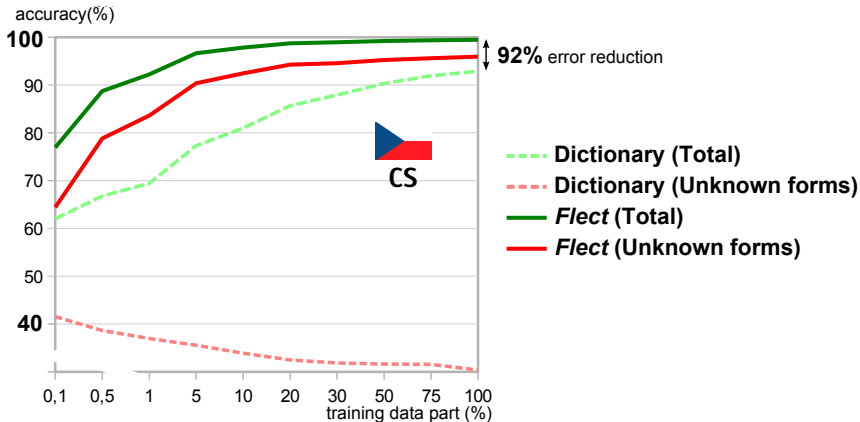
Flect vs. a dictionary from the same data

- 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



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/>

<http://bit.ly/flect>

The system is based on Python and Scikit-Learn.

You may contact us:

Ondřej Dušek & Filip Jurčiček

Charles University in Prague

odusek@ufal.mff.cuni.cz