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@-------------
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)

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)

Dolnokrčská X@-------------
artroplastika X@-------------

"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

Past Tense	Past Participle
grew	grown
flew	?

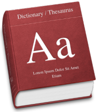
$$\frac{grew}{grown} = \frac{flew}{x}$$

$$x = \frac{fl\cancel{ew} \cdot \cancel{gr}own}{\cancel{gr}\cancel{ew}} = flown$$

Possible approaches to morphology

Dictionaries?

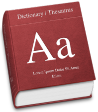
- Work well, reliable
- Limited coverage and/or availability



Possible approaches to morphology

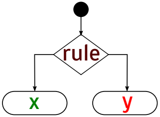
Dictionaries?

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



Hand-written rules?

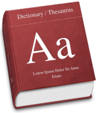
- Hard to maintain with complex morphology



Possible approaches to morphology

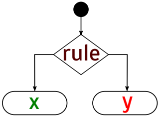
Dictionaries?

- 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

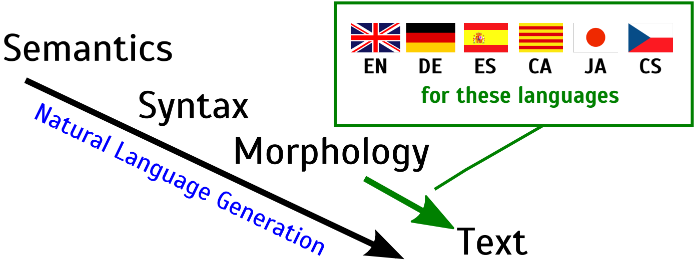
- Using machine learning to predict inflection

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

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

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

The task at hand

word + NNS → words

Wort + NN Neut,Pl,Dat → Wörtern

be + VBZ → is

ser + V gen=c,num=s,person=3, mood=indicative,tense=present → es

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

Casting inflection patterns as multi-class classification

[at the end]
[delete one letter]

fly
flies

>1-ies

[and add these]

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

[at the end]
[delete one letter]

fly
flies

>1-ies

[and add these]

sparen
gespart

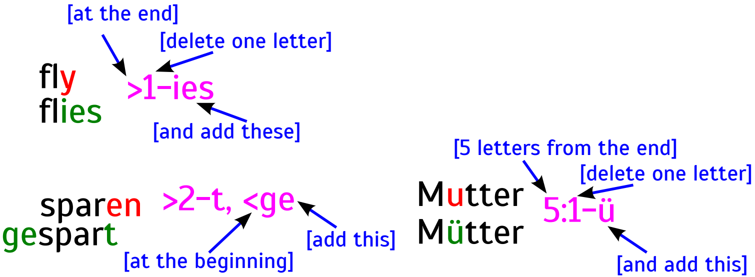
>2-t, <ge

[add this]
[at the beginning]

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

[at the end] [delete one letter]

fly
flies
>1–ies

[and add these]

[replace the whole word]

be
is
*is

sparen
gespart
>2–t, <ge

[at the beginning] [add this]

[5 letters from the end] [delete one letter]

Mutter
Mütter
5:1–ü

[and add this]

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 / fly + NNS → -ies

bind / find + VBD → -ound

Features useful for morphology generation

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

sky
fly + NNS → -ies

bind
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, fly	+ NNS	→	-ies
bind, 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
Pl
Neut
Dat

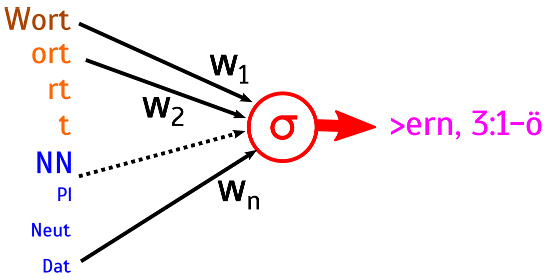
Our system *Flect*: Overall procedure

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

Wort
ort
rt
t
NN
Pl
Neut
Dat

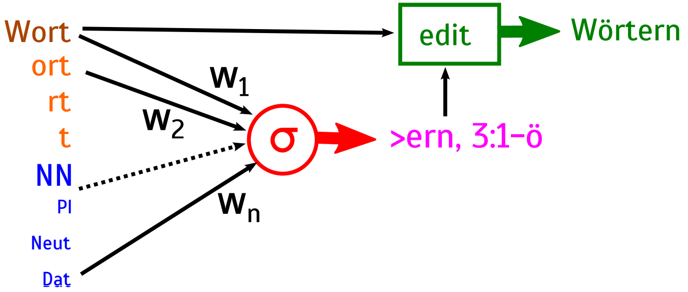
Our system *Flect*: Overall procedure

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

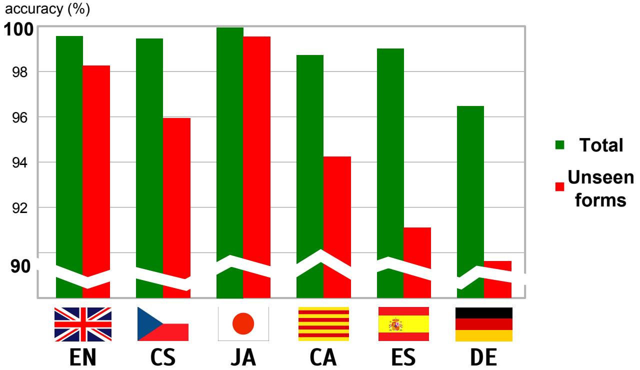


Testing *Flect* on 6 languages

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

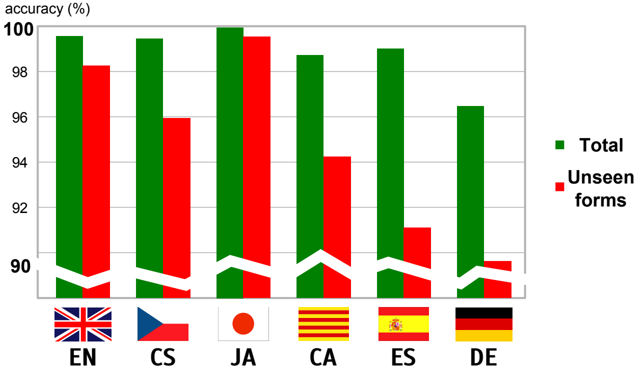
Testing *Flect* on 6 languages

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



Testing *Flect* on 6 languages

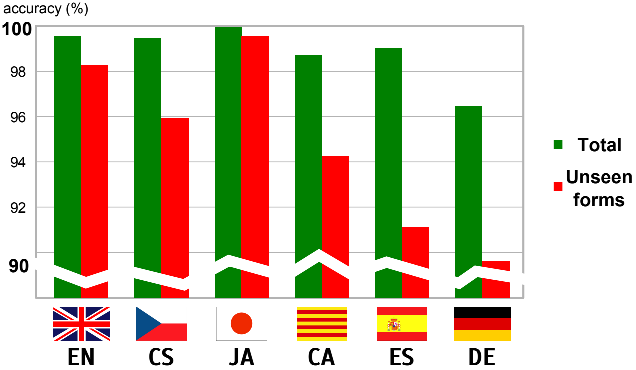
- **CoNLL 2009 data**: varying morphology richness & tagsets



- Works well even on unseen forms: suffixes help

Testing *Flect* on 6 languages

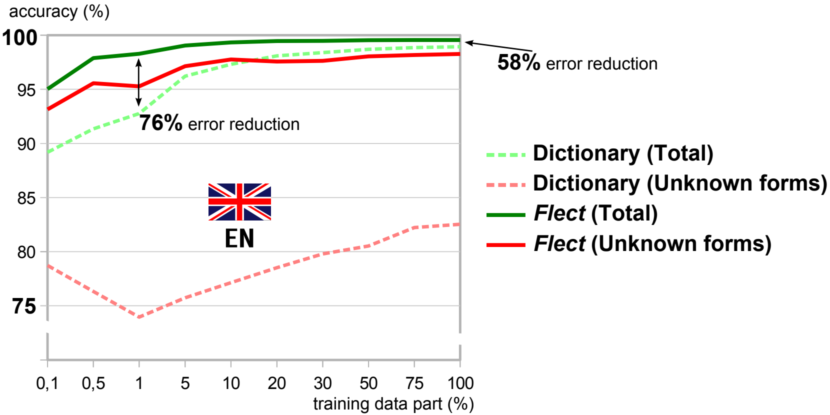
- **CoNLL 2009 data**: varying morphology richness & tagsets



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

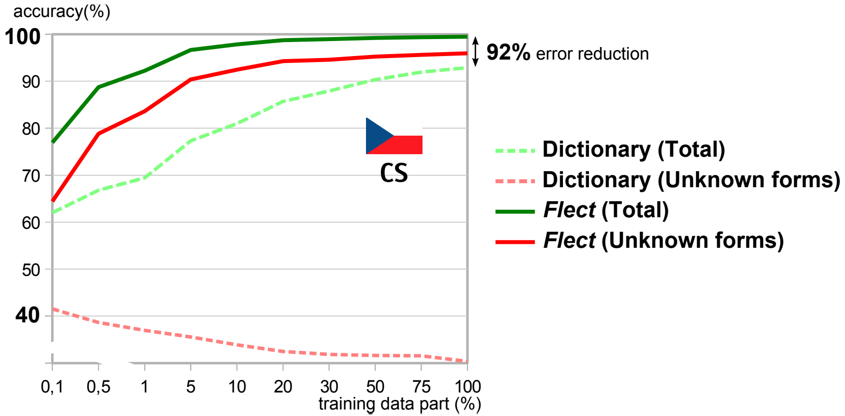
Flect vs. a dictionary from the same data

- English: Dictionary gets OK relatively soon



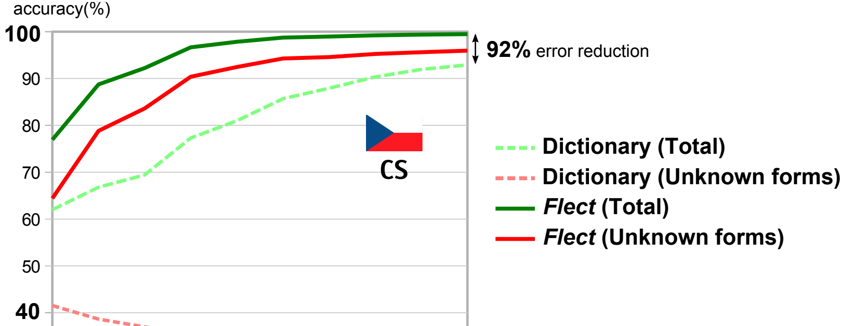
Flect vs. a dictionary from the same data

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



Flect vs. a dictionary from the same data

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



Dict	Hajič	*Flect*
92.88	98.25	99.45

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

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:

ženu

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

ženu	žena	NNFS4-----A----
	hnát	VB-S---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

ženu	žena	NNFS4-----A----	✓
	hnát	VB-S---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

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_;R_^(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-2_;R_^(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)

nejhezčímu >4–ký, <nej
hezký
[replace ending] [remove beginning]

- Not so new – some of the previous systems:
    - *Hajič (2004)*: statistical guesser (for forms that are not in the dictionary)
    - *Chrupała et al. (2008) – 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

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

- 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, occurence count threshold

Results: Morphological analysis

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

	cov (%)	**ø sugg.**
Hajič (060406)	98.82	3.85
Hajič (060406) + guesser	99.35	4.06
Hajič (131023)	98.52	4.00
Hajič (131023) + guesser	99.01	4.18
Memo-Suffixes (len 4)	98.71	5.69
Memo-Suffixes (len 3)	99.30	11.83
Memo-Suffixes (len 4, thr 2)	98.07	4.75
Memo-Suffixes (len 3, thr 2)	98.91	9.27

Results: Morphological analysis

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

	cov (%)	ø sugg.
Hajič (060406)	98.82	3.85
Hajič (060406) + guesser	99.35	4.06
Hajič (131023)	98.52	4.00
Hajič (131023) + guesser	99.01	4.18
Memo-Suffixes (len 4)	98.71	5.69
Memo-Suffixes (len 3)	99.30	11.83
Memo-Suffixes (len 4, thr 2)	98.07	4.75
Memo-Suffixes (len 3, thr 2)	98.91	9.27

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 %).

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

Results: Tagging

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

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

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*

Hajič, J. (2004). *Disambiguation of rich inflection: Computational morphology of Czech.* Karolinum.

The *Flect* generator is available for download:

`http://bit.ly/flect`

Contact me:

`odusek@ufal.mff.cuni.cz`, office 424