Resource-light Acquisition of Inflectional Paradigms

Radoslav Klíč and Jirka Hana

Geneea Analytics / MFF UK Praha

Outline

Introduction

- Morphology analysis necessary (IR etc.) but manual approach is expensive.
- We assume limited resources: plain text corpus and consultation with a native speaker.
- Our approach: modification and extension of Paramor, an unsupervised paradigm learner.
- We try to 'nudge' it towards correct analysis

Paradigms

- Classical Czech paradigms have slots for all combinations of relevant morphological categories.

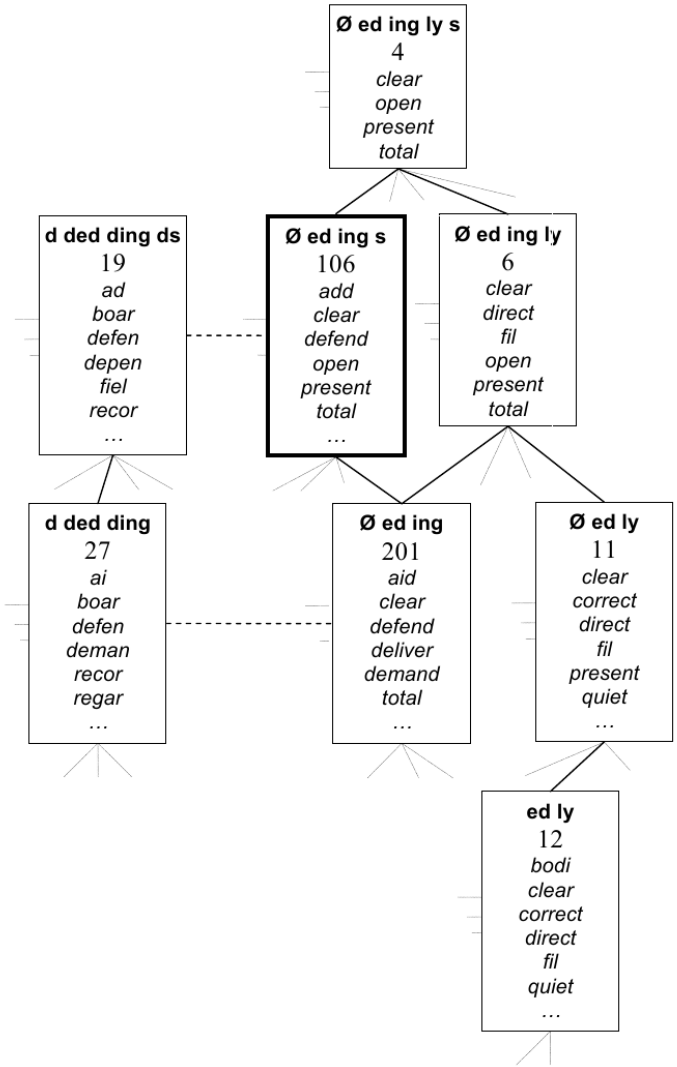
Case	Singular	Plural
nom	mat**k**+a	mat**k**+y
gen	mat**k**+y	mat**ek**+0
dat	mat**c**+e	mat**k**+ám
acc	mat**k**+u	mat**k**+y
voc	mat**k**+o	mat**k**+y
loc	mat**c**+e	mat**k**+ách
inst	mat**k**+ou	mat**k**+ami

- Low knowledge of grammar of the given language → simplified paradigms defined as a set of suffixes + set of stems e.g., (*a*, *y*, *e*, *u*, *o*, *ou*, *0*, *ám*, *ách*, *ami*) + (*žen*, *matk*, . . .)

Paramor – Schemes

- All splits of all words in the corpus → *c-stems* and *c-suffixes*
- In Paramor, partial paradigms are modelled by *schemes*.
- Scheme defined by a set of c-suffixes e.g., (*0*, *ed*, *ing*, *s*).
- scheme's stem set → all c-stems which form a word in the corpus with all the scheme's suffixes.
- Observation: adding a suffix cannot increase the number of scheme's adherent stems.

Scheme lattice



Paramor algorithm

- Bottom-up search. Starts with single-suffix schemes and ascends the lattice. Stops when the c-stem ratio drops below 0.25.
- Scheme clustering. Join similar schemes into scheme clusters. Similarity $\rightarrow$ similarity of (stem, suffix) pair sets. For example, schemes (*0*, *ly*, *ness*) and (*0*, *ly*, *er*, *est*) could be merged, as they share a lot of stem-suffix pairs like *deep* + *0*, *deep* + *ly*.
- Scheme cluster pruning.

Seeding

- Modified Paramor accepts manually entered input in the form of inflected word forms with marked morpheme boundary.
- Seed example: *matk/matc/matek + a, u, y / e / 0*
- Usage:
  - Add two-suffix schemes to the initial scheme set for bottom-up search.
  - Protect some scheme clusters from discarding.
  - Induction of allomorphy rules.

Allomorphy

- Allomorphy – more surface variants of a stem.
- Problem: try adding -*e* suffix to a scheme *(matk, noh)* + *(a, y, u, ou)*
- Oh no! neither *matke* nor *nohe* is in the corpus. (Although *matce* and *noze* are)
- Leads to
  - Incomplete schemes
  - Schemes with shifted morpheme boundary. *(mat)* + *(ka, ky, ce)*

Allomorphy – usage of the seed

- Induction of stem equivalence rules.
- From a seed entry

  $$politik/politic + a,\ u,\ ovi,\ em,\ y,\ \mathring{u},\ \mathring{u}m\ /\ i,\ ích$$

  we generate:

  $$*k \leftrightarrow *c\ /\ \{a,\ u,\ ovi,\ em,\ y,\ \mathring{u},\ \mathring{u}m\},\ \{i,\ ích\}$$

Evaluation

- Most common 'gold' data: lemmatised corpora
- For each word pair (w_1, w_2)
  - Do w_1 and w_2 belong to the same lemma?
  - Is there a scheme cluster generating both w_1 and w_2?
- Count true/false positives/negatives $\rightarrow$ precision and recall $\rightarrow$ F-score.

Results

F-score obtained with and without seeding:

Corpus	no seed	seed
cz	69.63	**72.99**
si	74.83	**75.61**
de	63.98	**64.52**
cat	62.74	**65.95**

Some problems

- Derivation vs inflection (*výuk* + *a*, *výuk* + *ový*)
- Prefixes (*ne-*, *nej-*)