CroMo - Morphological Analysis for Croatian

Damir Ćavar¹, Ivo-Pavao Jazbec² and Tomislav Stojanov²

Linguistics Department, University of Zadar¹
Institute of Croatian Language and Linguistics²

FSMNLP 2008
Scenario

- **Synchronic and diachronic study of language change and acquisition models**
  - Language data from a long period of time, and three major dialects in Croatia implying:
    - Variation wrt. e.g. string-based morphology or feature bundles
    - Ongoing discovery wrt. string combinatorics and features

- **Research questions require quantitative and qualitative information:**
  - of phonological, morphological, syntactic and semantic tokens and feature bundles, and their correlation and variation at various stages over time
Morphological segmentation and annotation
and lemmatization, and . . .

- Segmenting words:
  - *isponapijali su se* “they got drunk a little bit to satisfaction”
  - *is – po – napija – li*

- Annotating segments:
  - aspect prefix – aspect prefix – from stem-lemma *napiti* – plural participle

- Extending the annotation:
  - to a certain saturation – a little bit – “get drunk” from root-lemma *piti* – past event
FSA Architecture

Mapping of morph-groups to DFSAs (Mealy or Moore machine):

- **CroMo**
- **D. Ćavar**

**Outline**
- Introduction
- Model
- Evaluation
- Comments
Mapping ambiguity on emission: emission tuple 1 to $n$
Label DFSAs with variable names
Use rules referring to variable names for modeling of morphotactic regularities:

verbAspectPrefs* . verbAtiRoots . verbInflSuf
Generating potentially cyclic DFSAs:
FSA Architecture

Ambiguity mapped on emission tuple:
Lemmatization as a rule:

- Rightmost root is the semantic head
- Root-lemma: generate canonical word-form from the right-most root
  \[ \text{neprijatelja} \rightarrow \text{ne} + \text{prijatelj} + a \rightarrow \text{NEG} + \text{N-root} + \text{ACC} \]
  “not friend” = “enemy” \( \not\equiv \) friend not compositional!
  but useful for semantic field analysis!
  root-lemma: \( \text{neprijatelja} \rightarrow \text{prijatelj} \)

- Stem/base-lemma: generate canonical word-form from the stem without inflectional suffixes
  base-lemma: \( \text{neprijatelja} \rightarrow \text{neprijatelj} \)
Lemmatization (Hack):
- emission of byte-offset for suffix-elimination
- pointer to suffix string

Clean solution:
Implementation

- C++ wrapper for final application
- Ragel code (automaton definition) generated from morpheme DBs and rules, with associated feature bundles (extended version of Ragel, $\geq$ V. 6.1) for handling ambiguity via introduction of multiple emission symbols = emission tuples
- Ragel generated C code (jump-code)
Implementation

- Emission (feature bundles): as one bit-vector
- Features mapped from the General Ontology for Linguistic Description (upper ontology)
  - possibility: reasoning over linguistic concepts and features
- Optimization: mapping of concepts and their relations on a compressed bit-vector, maintaining inheritance and implicatures

![Diagram showing top-node concept, sub-class, and terminal-classes]
Evaluation

- **Hardware**: dual core 2.4 GHz
- **Lexical base**: 120,000 morphemes (and allomorphs)
- **Speed**: approx. 50,000 tokens per second with average morpheme count of 2.5 per token
- **Size**: binary footprint approx. 5 MB
- **Compilation (tables → Ragel + C; Ragel → C + DOT; gcc → bin)**: approx. 5 minutes, min. 4 GB RAM for monolithic architecture
Interoperability issues addressed:

- GOLD
- platform independent code
- code-page independence

Extensible (turnaround time of some minutes)

Minimally invasive and minimalistic

Open-source