The use of Extract Morphology for Automatic Generation of Language Technology for Votic

Kristian Kankainen

University of Tartu

January 7, 2019

Overview

What

- Source code generation extension to Språkbanken's *Morphology Lab* (Karp)
- encodes the lexical resource using Lexical Markup Framework
- resource is used as central description for code generators
  - for Grammatical Framework morphology module
  - for Giellatekno infrastructure integration

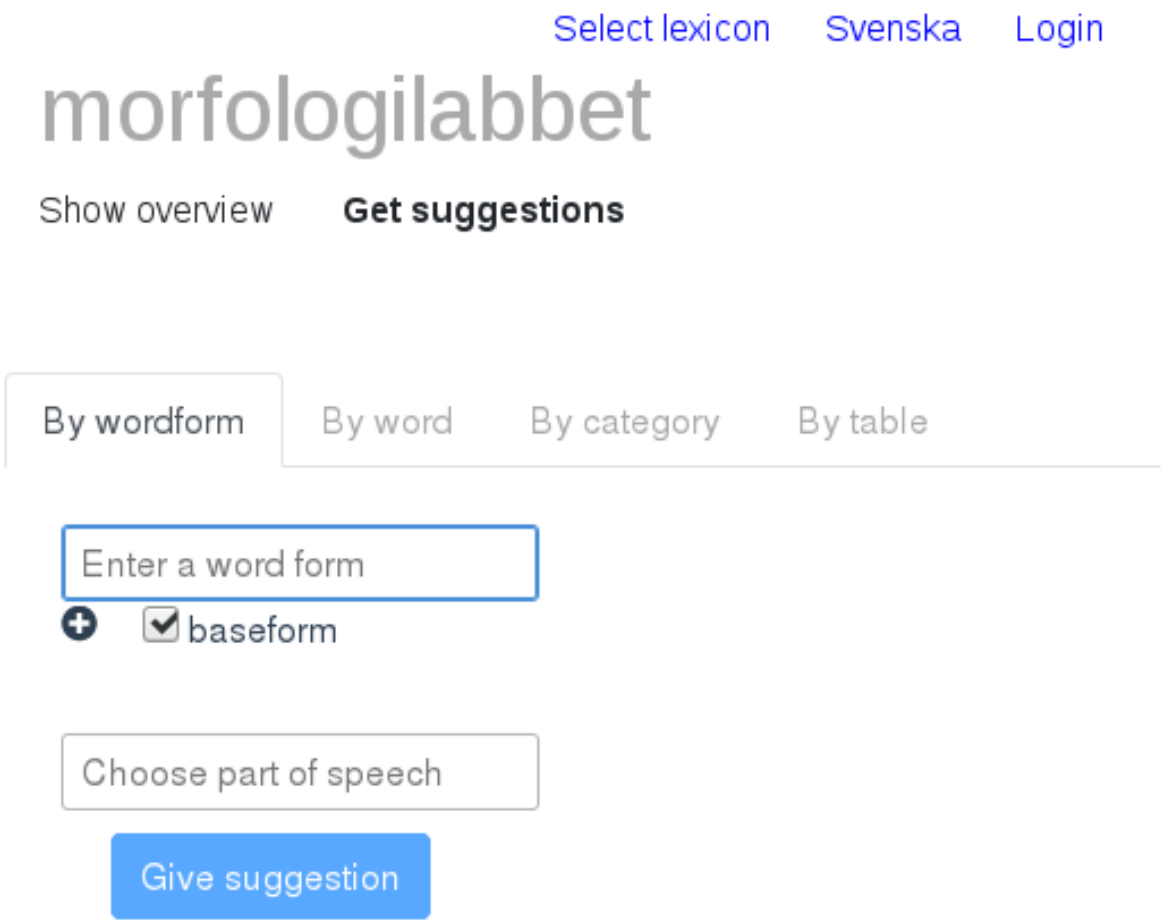
Why

- Puts the morphological resource in focus instead of lang tech
- updates in one single place
- more generators can be added
- computational morphology is defined by inflection tables, not programming
- tech neutral inflection tables is language documentation

The Votic situation in *Ämmesse vunukassaa*

- summer school *Ämmesse vunukassaa*
  - run by Heinike Heinsoo since 2009
  - appr 10 passive speakers
  - more children attending
- part of Votic corpus planning
  - keyboard layout for orthography
  - normative morphology (Vaipooli dialect)
  - modernization of lexicon
- main resources
  - dictionaries
    - "modern" dictionary (Heinsoo 2015, 5 languages, 1k entries)
    - Vaďďaa tšeelee sõna-tširja (2013, vot-est-rus, 30k entries)
  - corpus
    - Heinsoo's text books (15k tokens)
    - dictionary example sentences (330k tokens, est-rus parallel)

Språkbanken's Morphology Lab (Karp)



Språkbanken's Morphology Lab

Språkbanken's Morphology Lab:

- web interface for morphological resources in Karp
- integrates with corpus searches in Korp

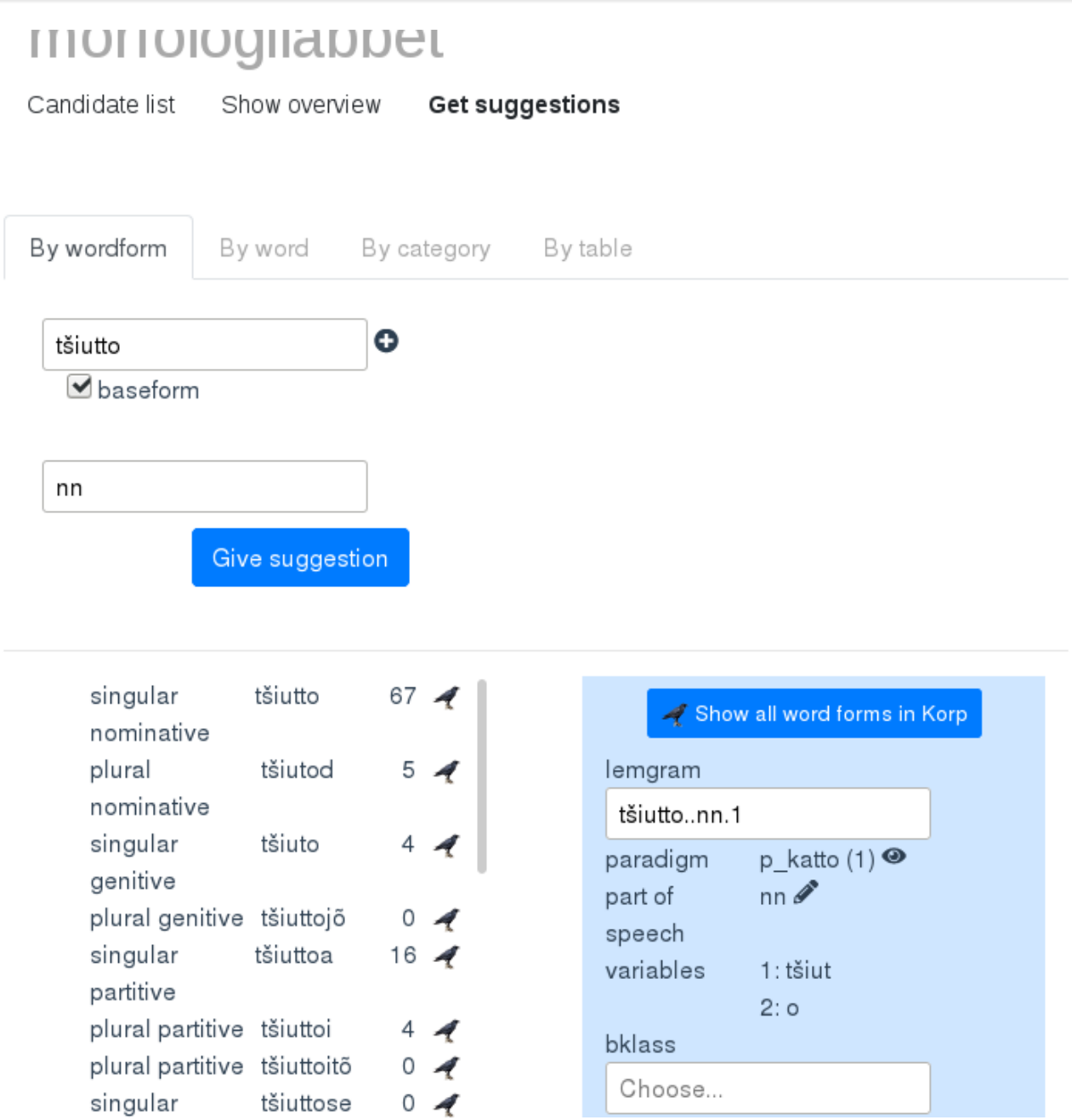
Workflow for adding new entries:

1. insert new word or wordform(s)
2. choose correct paradigm
   - by another word (shared paradigm word)
   - by inspecting generated inflection table
3. if no correct paradigm exists, insert inflection table

Workflow for changing entries:

- change wordform in inflection table
- paradigm updates automatically

Språkbanken's Morphology Lab



Extract Morphology in a nutshell

The inflection tables:

Wordform	MSD	LCS
katto	Sg Nom	kat t o
katod	Pl Nom	kat o d

Wordform	MSD	LCS
tšiutto	Sg Nom	tšiut t o
tšiutod	Pl Nom	tšiut o d

Extract Morphology in a nutshell

The inflection tables:

Wordform	MSD	LCS
katto	Sg Nom	kat t o
katod	Pl Nom	kat o d

Wordform	MSD	LCS
tšiutto	Sg Nom	tšiut t o
tšiutod	Pl Nom	tšiut o d

share the same abstract pattern:

Pattern	MSD
$x_1 \oplus \mathbf{t} \oplus x_2$	Sg Nom
$x_1 \oplus x_2 \oplus \mathbf{d}$	Pl Nom

with instantiations
$x_1 = kat$, $x_2 = o$ and
$x_1 = tšiut$, $x_2 = o$

Source code generation

- is an automatic process that generates source code from a central description
  - lexical resource is the central description (describes *what*)
  - source code generators implement the specifics (describes *how*)
- ISO Lexical Markup Framework used for central description
  - holds all information of the Karp lexical resource
  - inflection tables modelled with LMF Morphology module
  - paradigms' procedural descriptions (i.e concatenation patterns) modelled with LMF Morphological Pattern module

Mapping Extract Morphology patterns to GF source code

```
resource MorphoVot = {

  param
  Number = singular | plural ;
  Case = nominative ;
  NForm = NF Number Case ;

  oper
  Noun : Type = {s : NForm => Str} ;

  ----------------------------------------------
  -- Start of Noun section
  ----------------------------------------------

  mkTšiutto : Str -> Noun = \tšiutto ->
  case tšiutto of {
    tšiut + "t" + o@(-(_+"t"+_)) => mkTšiuttoConcrete tšiut o ;
    _ => Predef.error "Unsuitable lemma for mkTšiutto"
  } ;

  mkTšiuttoConcrete : Str -> Str -> Noun = \tšiut,o ->
  { s =
    table {
      NF singular nominative => tšiut + "t" + o ;
      NF plural nominative => tšiut + o + "d"
    }
  } ;
}
```

Integrating with Giellatekno

The LMF resource has everything to integrate with Giellatekno.
Things done:

- generate the lexicon
- generate FST source code for paradigms
- generate automatic tests

Things left to do:

- manually create a Makefile
- add interlingua pivots (Finnish translations)

Motivating benefits for integration:

- spell-checker
- bootstrap the Votic FST morphology tools
- pivot dictionary

Discussion

Main implications of the presented approach:

- the work on morphology is reduced to inflection tables
- the lexical resource is put in the middle
- the general benefits of source code generation
- benefits of lemma form agnosticity inherent in the approach

Reduced interface for working on computational morphology

- morphology defined by inflection tables, not source code
- knowledge of *wordforms*, not *morphology* or *programming*
  - enables the language community to do much work
- division of labor
  - most work done in the lexical resource
  - the language technologist can focus on programming or syntax
- the language technologist can leave the field
  - without abrupting the work

Lexical resource is put in the middle

- *what* instead of *how* gives technical neutrality
- LMF resource is (more) human readable than source code
  - value in longevity: 50 year old dictionary vs 50 year old source code
  - linguistical documentational value
- extendable with more generators
- updates in one central place
  - in the resource
  - in the source code

Benefits of source code generation

- extendable with more generators
  - resource updates propagated to all generated source codes
  - generated source code has coherent style
  - adding new technology doesn't outdate old technology
- updates in one central place (also for source code)
  - enables the lang technologist to re-write generators
- generators can use different terminology
  - *nominative, nom, Nom, etc*
  - specified by target

Benefits of lemma form agnosticity

```
tšiutto_N = mkKatto "tšiutto"
```

is a generated shorthand for

```
mkKattoConcrete "tšiut" "o"
```

- Extract Morphology doesn't depend on a lemma
- good for languages that lack a lexicographic tradition
- choice can be post-poned
  - permit more time to investigation
  - consult the dictionary users
  - lemma form can be chosen individually for each generator

More further work ...

- paradigm types, but subtypes?
  - *katto* vs *tüttö*
- pedagogically relevant for language revitalization
- grouping intentionally vs manual extensional grouping

Illustrated take home message

Illustrated take home message, if time permits.

Thank you!

Suurõd passibõd!