Morphology and Big Parse Charts in LKB/ACE

Olga Zamaraeva

For DELPH-IN summit

June 2016, Stanford, CA

Morphological grammars

- Precision grammars capable of analyzing/generating words consisting of morphemes
  - *walk+ing*
- Morphology section in the choices files
  - specification
- Morphological rules (e.g. irules) in the TDL files
  - Implementation in precision grammars

Morphological grammars

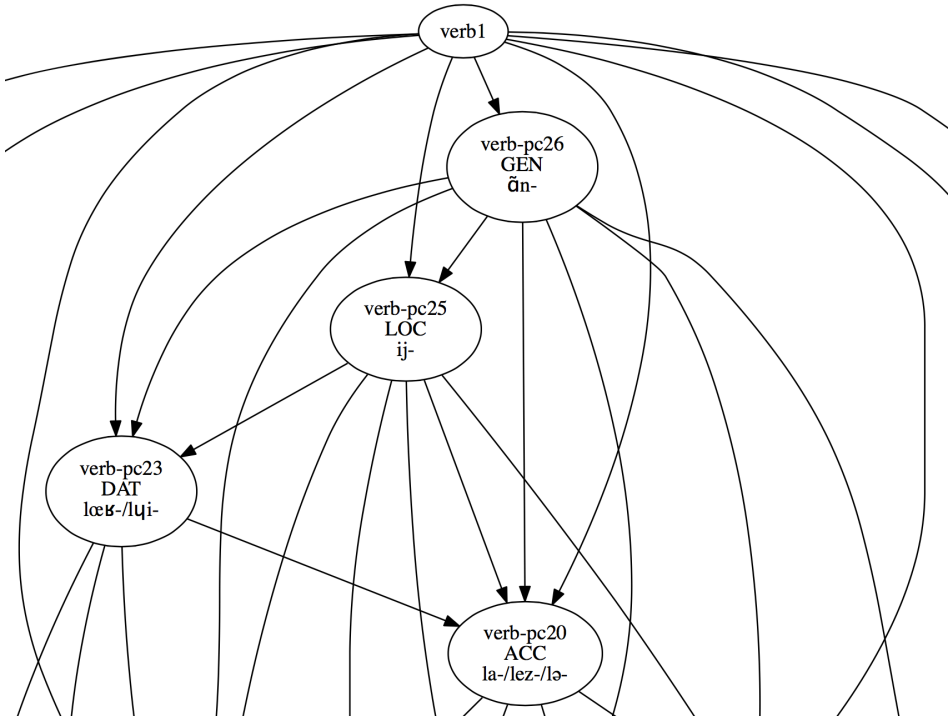
- Precision grammars capable of analyzing/generating words consisting of morphemes
  - *walk+ing*
- **Morphology section in the choices files**
  - **specification**
- Morphological rules (e.g. irules) in the TDL files
  - Implementation in precision grammars

Morphological grammars: choices

```
verb-slot1_order=after
  verb-slot1_input1_type=verb
  verb-slot1_morph1_name=prp
  verb-slot1_morph1_orth=ing
    verb-slot1_morph1_feat1_name=aspect
    verb-slot1_morph1_feat1_value=prog
    verb-slot1_morph1_feat1_head=verb
    verb-slot1_morph1_feat2_name=form
    verb-slot1_morph1_feat2_value=prp
```

Affix graph

- Affixes are nodes
- Edges are input relations



Goal: Infer Morphology Automatically

- Field linguists do not have time to go through all their data by hand
- A system which offers them hypotheses would be helpful
  - Position classes candidates
  - Affixes participating in circumfixation
  - Etc.

Chintang [ctn]

- IGT collection (Bickel et al., 2013)
- Polysynthetic language
- Possibly variable affix order
- Circumfixes
- Possibly iterating affixes

Oracle grammar (Bender et al. 2012)

- 54 verb position classes (+ stems)
- 54 edges (just one input for each position class in the graph).

Automatically Inferred Grammars

- MOM (Wax, 2014)
  - 54 position classes (input overlap = 30%)
  - ~400 edges

- Clustered affixes (Zamaraeva, in press)
  - 54/58 position classes (k=54)
  - ~800 edges

Evaluation by Morphological Parsing

- Extract words from test sentences
- Set argument optionality in the grammar so that anything can be dropped
- Run the grammar on the test words with LKB/ACE
- Evaluate
  - Also, using [incr tsdb()]

LKB/ACE technical issues

- Chart parser
- LKB: Max number of rules to try to apply
  - 4K default
  - ~25K possible before 2GB memory is used up
  - 32-bit Common Lisp license (at UW)
- ACE:
  - Similar story?.. Will skip the item if it requires too much RAM
- [incr tsdb()] in-built in the LKB fails with an out of memory error

Evaluation: failures

- True failures (no path in graph)
  - *yuŋ-ma-dis-ma*
    - NOTE: lexemes do not span position 0 `yuŋ-ma-dis-ma'!
    - NOTE: post reduction gap
    - SKIP: yuŋ-ma-dis-ma
- Technical failures (too many paths in graph)
  - *lond-a-ce-a-ŋ-e*
    - NOTE: terminating search, too much RAM
    - SKIP: lond-a-ce-a-ŋ-e

What to do?..

- We want to be able to infer morphotactics automatically (or do we?)
  - Impose a limit on possible paths?
    - Weigh paths and discard some?
- We want to be able to evaluate them by parsing
  - Are any improvements to the parsing/testing software possible/realistic in the near future?