Knowledge Engineering for NLP

April 10, 2006

The Grammar Matrix

Overview

- Goals
- Architecture
- Modules: State of the art
- Technical details

Grammar Matrix: Goals

- Speed-up precision grammar development
- Standardize semantic representations
- Bottom-up exploration of linguistic universals

Grammar Matrix: Architecture

- Universally useful contraints: matrix core
- Recurring by non universal patterns: modules
  - (Suggestions for a better term?)
- Matrix configuaration cgi script takes relevant info from modules and adds it to my_language.tdl and instance files.
- User builds out my_language.tdl and instance files

Grammar Matrix Modules: Workflow

- Pick phenomenon
- Do typological study and/or write lab instructions for 567
- Abstract from above tdl for various realizations
- Incorporate into matrix configuration script

Grammar Matrix Modules: State of the art

- Word order
- Sentential negation
- Coordination
- Matrix yes-no questions
- Hidden in the lexicon...

Word order

- All six strict orders
- V-initial, V-final
- noun-det order
- adposition order
- Coming soon? V2
- Longer term: splattered NPs; subordinate clauses

Sentential negation

- Multiple strategies anticipated
- Multiple strategies within a single language allowed
- Not yet based on typological literature
- Some strategies may be missing

Coordination (Drellishak)

- Based on a typological study
- ’and’-type coordination (with unbounded numbers of coordinands)
- Multiple (exhaustive?) strategies implemented
- Multiple strategies within a single language allowed
- Interaction with e.g., case or agreement left to grammar developer
- Not yet: with-type coordination, ’both ... and’, or, but

Matrix yes-no questions

- A few strategies implemented
- No typological study yet
- NO SEMANTICS

Hidden in the lexicon

- NP v. PP arguments for simple (in)transitives
- Some of the word order stuff (adpositions)
- A few types of auxiliaries
- A tiny number of subcategorization types

More on valence patterns

- Extensive support within the matrix core
- Long-term modules support awaits work on:
  - Supporting user definitions of lexical types (e.g., quirky case)
  - Interfaces for importing lexicons created with other tools (FIELD)
  - Better UI for interactive addition of lexical items
  - Morphophonology...

Other notes

- Current configuration script assumes full-form lexicon
- Matrix + modules do morphosyntax, syntax, syntax-semantics interface
- We'll always be assuming regularized morphology
- Need to find a way to support building morphophonological analyzers & interface with them

Long term vision

- Detailed typological questionnaire outputs good approximation precision grammar
- Lexical input mechanism dovetails nicely with dictionary construction
- Infrastructure for field linguist-grammar engineer-native speaker collaboration
- Ontological annotation of tdl
- Automated lexical acquisition from corpora?

Evaluation

- Do the modules as implemented in fact cover the grammatical systems they purport to? (Poulson 2006)
- Do the modules as implemented interact properly? (Poulson 2006)
- How much do the Matrix+modules speed up grammar development?
- How easy is it to work with/expand the Matrix?
- What is the coverage of the Matrix in terms of the world's languages?

Technical details

- Table of contents (files and what they contain)
- Feature geometry
- Tour of types defined in the matrix (begin)

Table of contents (1/3)

Table of contents (2/3)

Table of contents (3/3)

On to the tour...

- Feature geometry FAQ
- matrix.tdl

Overview

- Goals
- Architecture
- Modules: State of the art
- Technical details