Knowledge Engineering for NLP

January 23, 2008

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
- Computational linguistic typology

Grammar Matrix: Architecture

- Universally useful contraints: matrix core
- Recurring by non universal patterns: libraries
- 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 Libraries: 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 Libraries: 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
- Now with 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: getting bigger as Scott adds the first version of the case functionality.

More on valence patterns

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

Other notes

- Until recently, configuration script assumed full-form lexicon; this is just starting to chnage
- Matrix + libraries 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 libraries as implemented in fact cover the grammatical systems they purport to? (Poulson 2006; Bender et al 2007)
- Do the libraries as implemented interact properly? (Poulson 2006; Bender et al 2007)
- How much do the Matrix+libraries speed up grammar development? (Wambaya test case)
- 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)

matrix.tdl	core types
head-types.tdl	more core types, too unreadable for matrix.tdl
my_langugee.tdl	language-specific types

Table of contents (2/3)

rules.tdl	phrase structure rule instances
irules.tdl	spelling changes lex rule instances
lrules.tdl	spelling preserving lex rule instances
lexicon.tdl	lexical entries
irregs.tab	table of irregular forms
roots.tdl	start symbols

Table of contents (3/3)

lkb/script	load file for lkb interaction
lkb/user-fns.lsp	grammar-specific lisp functions
lkb/globals.lsp	grammar-specific global variables
	(includes punctuation characters to strip)
lkb/mrsglobals.lsp	grammar-specific global variables
	for processing of semantic representations
(lkb/mt.lsp	global variables used by transfer code)
(lkb/transfer.tdl	baby transfer grammar)

On to the tour...

- Feature geometry FAQ
- matrix.tdl

Overview

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