Knowledge Engineering for NLP January 21, 2009 The Grammar Matrix **Overview**

- Goals
- Architecture
- 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 configuration 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
- Design additions to questionnaire and associated validation
- Modify customization script to emit correct tdl per questionnaire responses
- Add unit tests

Other notes

- Until recently, configuration script assumed full-form lexicon; this is no longer so
- 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 or external resources

Evaluation

- Do the libraries as implemented in fact cover the grammatical systems they purport to? (Poulson 2006; Bender et al 2007; Bender et al in proress)
- Do the libraries as implemented interact properly? (Poulson 2006; Bender et al 2007)
- How much do the Matrix+libraries speed up grammar development? (Bender 2008)
- 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 typeshead-types.tdl more core types, too unreadable for matrix.tdlmy_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
- MRS appreciation

Overview

- Goals
- Architecture
- Technical details