Grammar Prototyping and Testing with the LinGO Grammar Matrix Customization System

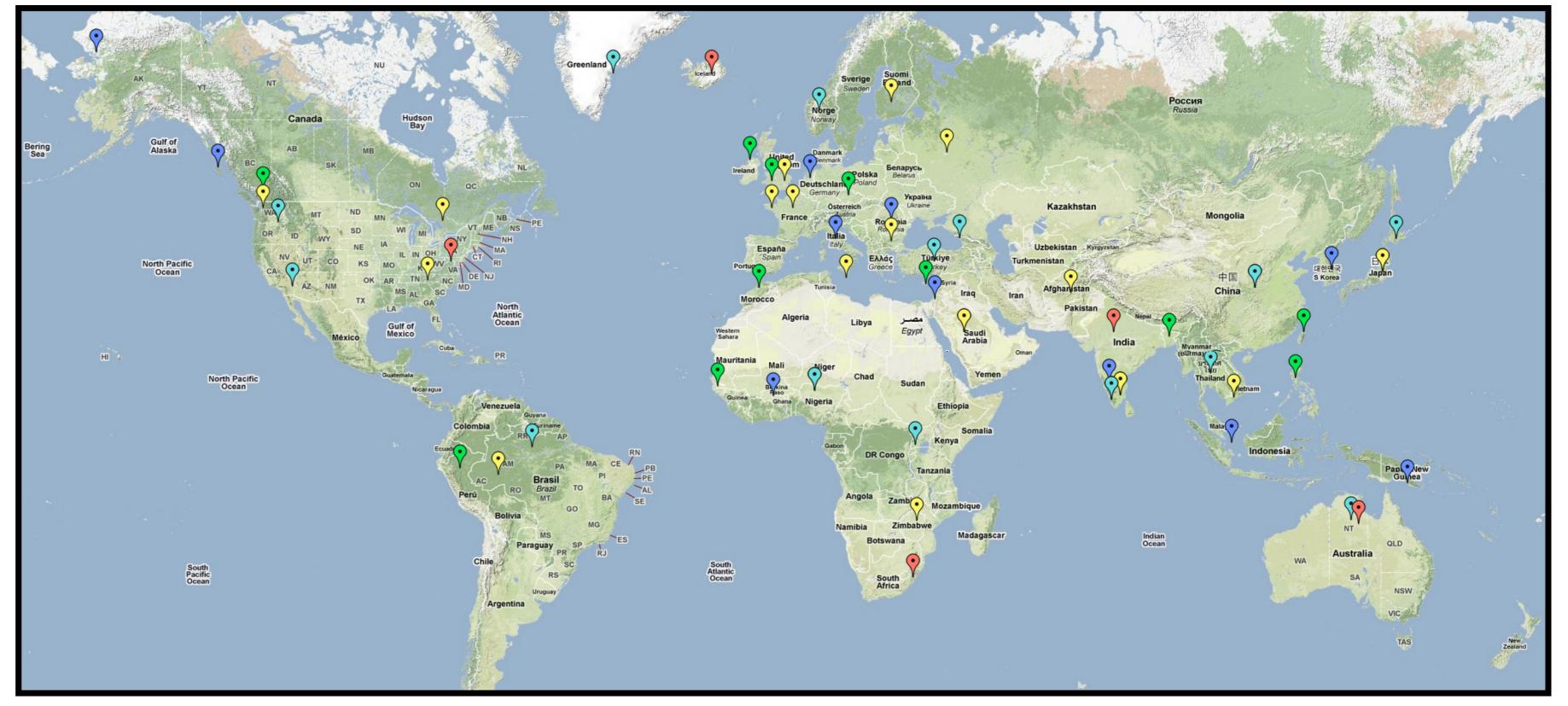**Emily M. Bender**◇, **Scott Drellishak**◇, **Antske Fokkens**♠, **Michael Wayne Goodman**◇, **Daniel P. Mills**◇, **Laurie Poulson**◇, **Safiyyah Saleem**◇

◇ University of Washington ♠ Saarland University

1. Background

The LinGO Grammar Matrix

- The LinGO Grammar Matrix customization system is a repository of distilled linguistic knowledge.
- This web-based service
  - elicits typological descriptions of languages from users
  - yields customized grammar fragments
- Grammar fragments are ready for sustained development into broad-coverage grammars.
- The grammar framework used is Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994).
- The grammars map strings to semantic representations in the format of Minimal Recursion Semantics (MRS; Copestake et al., 2005).

2. Grammar Customization

System Design

- Presents users with options for linguistic phenomena.
- Form values entered are stored in a 'choices file'.
- Forms dynamically update based on current inputs.
- Validation helps prevent incorrect or unlikely grammars.
- A validated choices file can be compiled into a TDL grammar (Copestake, 2002), then downloaded for further development by hand, or tested in the browser.

Linguistic Phenomena Covered

- Word order, case, direct-inverse marking, coordination, argument optionality, matrix yes/no questions
- Person, number, gender
- Tense and aspect, negation, custom features hierarchies
- Lexicon (nouns, transitive and intransitive verbs, auxiliaries, determiners, adpositions), lexical rules

3. Test by Generation

Testing in the browser

- A validated grammar may be loaded on the server and used to generate sentences from semantic templates.
- Generated sentences allow in-browser exploration of consequences of analytical choices.
- Leverages generation algorithm (Carroll et al., 1999) in the LKB (Copestake, 2002).
- Allows identification of over- and under-generation.
- Users can interactively explore parse trees and semantic representations for realized strings.

Underspecified MRS Templates

- Stored MRS templates are generic enough to be used in any language (see Figure 2).
- Templates are specialized according to the choices file.
- Users can control space of generation by constraining semantic features in the input MRSs.
- Initial results are limited to one semantic predicate per verb type, but users can choose to see more.

Languages of grammars created with the customization system. Map image courtesy of Google Maps, location data courtesy of WALS.

Figure 1: *Test-by-generation screenshot.*

4. Customized Grammars

Grammar Statistics

- As the customization system expands to include analyses of more phenomena, the customized grammar it outputs grow in complexity.
- The test by generation methodology is intended to help users understand the behavior of their grammars and thereby get the most out of the customization system.
- In Figure 3, we compare 11 grammars customized in a graduate course to a broad coverage hand-developed grammar (the ERG; Flickinger, 2000).
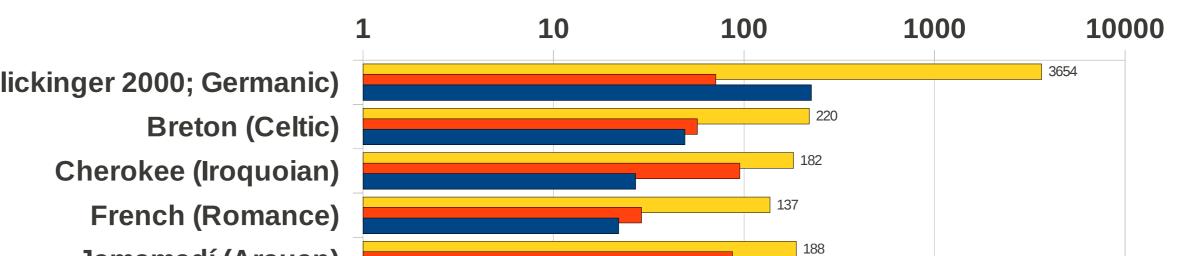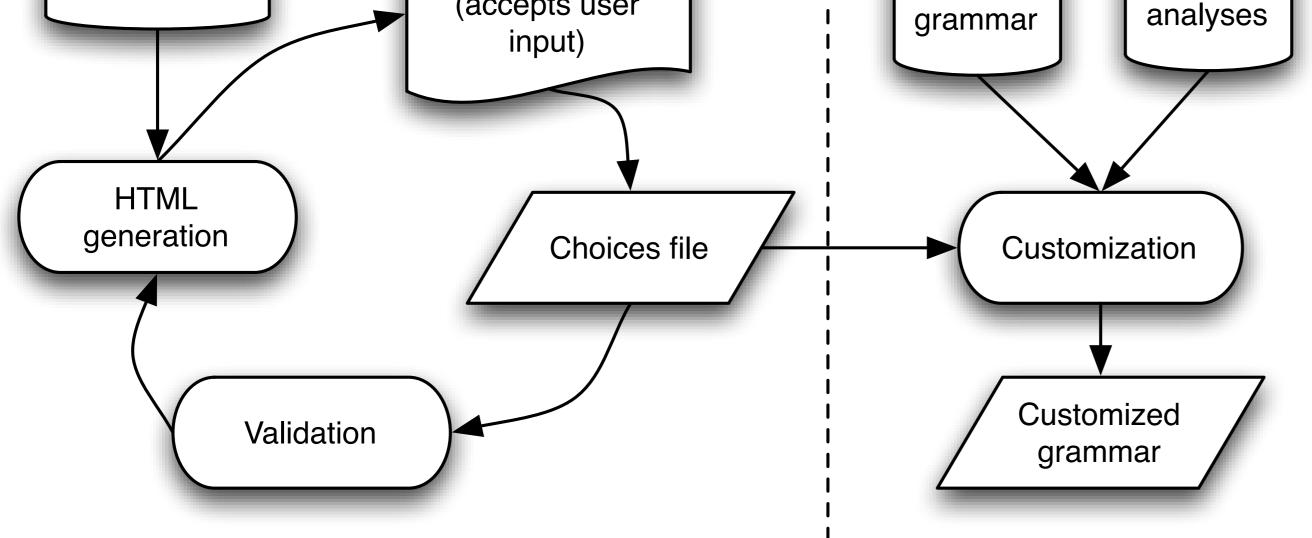
5. Availability

`http://www.delph-in.net/matrix/customize`

a. ⟨ h1,e2, {h7:_cat_n_rel(x4:SG:THIRD), h3:exist_q_rel(x4, h5, h6), h1:_sleep_v_rel(e2:PRES, x4)}, {h5 qeq h7} ⟩

b. ⟨ h1,e2, {h7:#NOUN1#(x4), h3:#DET1#(x4, h5, h6), h1:#VERB#(e2, x4)}, {h5 qeq h7} ⟩

Figure 2: *English MRS (a) and MRS template (b)*

6. Acknowledgments

References

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proc. of the Workshop on Grammar Engineering and Evaluation at COLING 2002*, pages 8–14.

John Carroll, Ann Copestake, Dan Flickinger, and Victor Poznański. 1999. An efficient chart generator for (semi-) lexicalist grammars. In *Proc. of the 7th European workshop on natural language generation (EWNLG99)*, pages 86–95.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI, Stanford.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language & Computation*, 3(4):281–332.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6:15−28.

Martin Haspelmath, Matthew S. Dryer, David Gil, and Bernard Comrie, editors. 2008. *The World Atlas of Language Structures Online*. Max Planck Digital Library, Munich. Http://wals.info.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, IL.

Elicitation of typological information — Grammar creation

Figure 3: *Schematic system overview*

Figure 4: *Grammar types, lexical rules and phrase structure rules for 11 customized grammars and the ERG*