The Grammar Matrix: Computational Syntax and Typology

Scott Drellishak
University of Washington
MPI EVA, 23.10.2006

Overview

- In this talk, I'll describe the Grammar Matrix, a project to develop a cross-linguistic foundation for computational syntax
- In particular, how we deal differently with (apparently) universal and non-universal but widespread phenomena
- First, a bit of background: what we mean by "computational syntax"

- Computational Syntax
- The Matrix
- Matrix Libraries
- Demo
- My Research
- The Matrix and Typology

Computational Syntax

- Detailed description of a language, entirely formalized—even a computer can do it
- In this project, formal system is HPSG (Pollard & Sag 1994, Sag et al. 2003) encoded in TDL format
- This allows our grammars to run in the freely-available LKB environment (Copestake 2002)
- This system can parse sentences to a semantic representation and also generate from that representation back to sentences

- Computational Syntax
- ➢ The Matrix
- Matrix Libraries
- Demo
- My Research
- The Matrix and Typology

What is the Matrix?

- Purpose: Distilling the wisdom of existing broad coverage grammars into a common foundation for computational syntax
- Initially based on:
  - English Resource Grammar (Flickinger 2000)
  - A Japanese grammar (Siegel & Bender 2002)
- Since then, extended and generalized through exposure to projects implementing grammars for other languages

What's in the Matrix?

- Basic HPSG feature definitions and technical devices (e.g. list manipulation)
- Types that support a semantic representation, Minimal Recursion Semantics (Copestake et al. 2001)
- Classes of grammatical rules: derivational and inflectional, unary and binary phrase structure, head-initial and head-final, head-complement, head-specifier, head-subject, etc.
- Simple part-of-speech inventory: verb, noun, adjective, adverb, adposition, complementizer, determiner, number-name, conjunction
- Follows general HPSG principles, e.g. semantic compositionality, phrases generally identified by heads

Implementing a Grammar

- Particular languages implemented by multiple inheritance from the appropriate Matrix rules
- Example: SV word order
- A language-specific subj-head rule inherits from two Matrix rules:
  - A basic-subj-head rule for the semantics
  - A head-final rule that specifies the order (note: assumes V is the head of S)

Grammars Implemented

- Emily Bender regularly teaches a grammar engineering class
- Each student picks a language and implements a grammar for it based on the Matrix
- These languages include:
  - Arabic, Akan, Armenian, Basque, Cantonese, Esperanto, Farsi, Finnish, French, Haitian Creole, Hawaiian, Hindi, Hungarian, Japanese, Latin, Mongolian, Navajo, Polish, Portuguese, Russian, Spanish, Swahili, Swedish, Tigrinya, Turkish, and Uzbek.

Is the Matrix Universal?

- Intended to contain what's shared among all languages
- …but not everything that's common is universal:
  - not all languages have the same inventory of parts of speech
  - coordination not in all languages
- What do we do with non-universal phenomena?

- Computational Syntax
- The Matrix
- ➢ Matrix Libraries
- Demo
- My Research
- The Matrix and Typology

Libraries

- Our solution for phenomena that are in many, but not all languages
- Some of these phenomena simply don’t occur in all languages (e.g. coordination)
- Others do, but the details of their expression differ (e.g. word order)
- Such phenomena are still necessary for a (possibly large) subset of grammar writers

Contents of a Library

- A Matrix library consists of three parts:
  - HPSG rules implementing a phenomenon
  - A web questionnaire that allows a grammar-writer to describe the phenomenon in the language in question
  - Software that takes the answers and creates a grammar
- Libraries should be as general as possible to cover as wide a range of typological variation as possible

Current Libraries

- Word Order: SOV, SVO, VSO, OSV, OVS, VOS, V-final, V-initial, free
- Sentential Negation: inflection on main or aux verb; adverb modifying S, VP, or V; or both
- Coordination: lexical or morphological marking, different patterns of marking, different phrase types covered
- Yes/No Questions: subj-verb inversion (main, aux, or both), question particle, intonation only

- Computational Syntax
- The Matrix
- Matrix Libraries
- ➢ Demo
- My Research
- The Matrix and Typology

- Computational Syntax
- The Matrix
- Matrix Libraries
- Demo
- ➢ My Research
- The Matrix and Typology

My Research

- Implementing libraries for phenomena we currently lack
  - Coordination (Drellishak & Bender 2005)—first version done, second version planned
  - Case (on nouns, for the time being)
  - Agreement between verbs and their arguments (entails support for at least person and number as well)

Coordination

- Strategies vary in four dimensions:
  - Kind of marking: lexical, morphological, none
  - Pattern: one marked: “A B and C” (monosyndetic), *n*-1 marked: “A and B and C” (polysyndetic), *n* marked: “and A and B and C” (“omnisyndetic”), none marked: “A B C” (asyndetic)
  - Position: before or after: “and A” or “A and”
  - Types of phrases covered
- (Some known strategies aren’t covered)

Case

- Currently, only case-marking adpositions supported (in the Lexicon section)
- For a fuller implementation, we need:
  - How case can be marked (affixes, adpositions, …)
  - What is marked (Only the noun? The whole noun phrase?)
  - Arguments marking patterns (ergativity)
  - A clean interface

Agreement

- Verbs agree with their arguments in various ways (e.g. person and number)
- To implement agreement, we need:
  - What can agree?
  - Which arguments agree?
  - How does agreement interact with case (especially ergativity)?
  - A clean interface

Dependencies

Proposed Library	Known Dependencies (transitive)
Case	
Gender (and noun classes generally)	
Person and Number	
Pronouns	Case, Gender, P&N
Agreement	Case, Gender, P&N
Adpositional Phrases	Case
Verb Classes	
Argument Optionality	Verb Classes
Long-distance Dependencies	Pronouns
Relative Clauses	Long-distance Dependencies
Content Questions	Long-distance Dependencies
Numeral Classifiers	P&N
Evidentiality	?
Noun Incorporation	Pronouns, ?

- Computational Syntax
- The Matrix
- Matrix Libraries
- Demo
- My Research
- ➢ The Matrix and Typology

Matrix Development

- Our immediate purpose is providing grammar-writers with a foundation
  - Includes grammar engineers, linguists describing languages, language preservation efforts…
  - We provide a starter grammar, they continue in as much detail as they like
  - The problems they encounter inform changes and improvements to the Matrix

Bottom-Up Typology

- This process gives us “bottom-up, data-driven investigation of linguistic universals and constraints on cross-linguistic variation” (Bender & Flickinger 2005)
- Formalizing grammars in a single framework exposes interesting similarities, differences, and issues:
  - In coordination, *n* marks different from *n*-1, because only *n*-1 binary semantic relations are needed for *n* coordinands
- We hope to “harvest” typological insights during the process of developing the Matrix

Future Development

- The Matrix is “applied linguistics”—practically, that means it’s never complete and will contain compromises
- Over time, the core Matrix will grow (probably slowly) as new generalizations are found
- “Universals” found not to be universal will tend to migrate out of the Matrix into libraries

Big Picture

- Every research project has *contributors* and an *intended audience*
  - e.g. the Matrix: we contribute an implementation, aimed at grammar writers
- With respect to typology, the Matrix is both
  - At the moment, we're consumers of the research output of typologists
  - In the longer term, we hope to contribute new knowledge to the field

References

Bender, Emily M. and Dan Flickinger. 2005. Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. Proceedings of IJCNLP-05 (Posters/Demos), Jeju Island, Korea.

Copestake, Ann, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 1999. Minimal Recursion Semantics: An introduction. Language and Computation, 1 (3): 1-47.

Copestake, Ann. 2002. Implementing Typed Feature Structure Grammars. Stanford, CA: CSLI Publications.

Drellishak, Scott and Emily M. Bender. 2005. A coordination module for a crosslinguistic grammar resource. In Proceedings of the HPSG05 Conference.

Flickinger, Dan. 2000. On building a more efficient grammar by exploiting types. Natural Language Engineering 6 (1) (Special issue on efficient processing with HPSG): 15-28.

Pollard, Carl and Ivan A. Sag. 1994. Head-Driven Phrase Structure Grammar. Chicago, IL and Stanford, CA: U. of Chicago Press and CSLI.

Sag, Ivan A., Thomas Wasow, and Emily M. Bender. 2003. Syntactic Theory: A Formal Introduction. Stanford, CA: CSLI Publications.

Siegel, Melanie and Emily M. Bender. 2002. Efficient deep processing of Japanese. In Proceedings of the 3rd Workshop on Asian Language Resources and Standardization at 19th International Conference on Computational Linguistics, Taipei, Taiwan.