Grammar Engineering for Crosslinguistic Hypothesis Testing

Emily M. Bender
Department of Linguistics
University of Washington
October 13, 2006

Acknolwedgments

- Dan Flickinger, Stephan Oepen
- Scott Drellishak, Laurie Poulson
- Students in Grammar Engineering classes (past 3 years)

Overview

- Big issue: Hypothesis testing in syntax
- Specific work: Grammar Matrix customization system

Road map

- Syntactic hypothesis testing
- Two classic observations
- Grammar engineering in general terms
- Some specifics about the Grammar Matrix project
- Conclusion and implications

Definitions

- Syntax: The means by which natural languages relate strings of words to their meanings, over an infinite set of possible strings of words
  - Secondarily: A system which models syntactic well-formedness
- Syntactic hypothesis: A hypothesis about the structures assigned to a class of sentences or more broadly about constraints on possible grammars

Syntactic hypotheses: Constraints on grammars

- P&P style UG
- Compositionality
- Movement vs. lack thereof
- Empty categories vs. lack thereof
- ‘Generative’ approach v. exemplar-based+analogy
- General rules and idiosyncrasies stored in the same system

Syntactic hypotheses: Types of structures

- Most constituents have heads
- Agreement is fundamentally both syntactic and semantic
- Case on nouns is determined by selecting heads
- Long-distance dependencies are mediated by local dependencies (‘looping’ rather than ‘swooping’ movement)

Syntactic hypotheses: Predictions about languages

- No languages mark coordination with a single conjunction at the beginning of a list of coordinands
- All languages have some way to express statements, commands, and questions
- No language allows the extraction of a coordinand (CSC: element constraint, Ross 1967)

Testing hypotheses

- Can’t just go look: these properties aren’t typically apparent in surface strings, nor are they accessible to introspection
- Instead: Build a model, and test its predictions about grammaticality against judgments of acceptability
  - Predictions about languages
  - Predictions within languages

Models

- Sketched: Argue that a model with(out) property X can't work
- Elaborated: Process test examples with the model and calculate predictions of grammaticality
  - Can include examples testing interaction with many parts of the grammar
  - Can include open corpus data, to catch examples of the phenomenon in question unanticipated by the linguist

Observation one

- Meillet (1903) [or possibly de Saussure or von der Gabelentz]:

  “que chaque langage forme un système où tout se tient”

  - For the structuralists: It’s all about the contrasts
  - For grammar engineers: It’s all about the interactions

Observation two

- Chomsky (1965)

  "To the extent that a linguistic theory succeeds in selecting a descriptively adequate grammar on the basis of primary linguistic data, we can say that it meets the condition of *explanatory adequacy*."

  - Explanatory adequacy presupposes descriptive adequacy.

Upshot

- It is not possible to test a syntactic hypothesis in one subdomain without simultaneously building a model of many intersecting subdomains.
- It is not possible to test a syntactic hypothesis without considering a wide variety of sentences, to illustrate the interaction of subdomains.

Observation two-prime

- Chomsky & Lasnik (1995)

  “Suppose we have a collection of phenomena in a particular language. [...] there are many potential rule systems, and it is often possible to devise one that will more or less work [...] But this achievement, however difficult, does not count as a real result if we adopt the P&P approach as a goal.”

  - How can we tell when we have a rule system that works?

Grammar Engineering

- Building models on a computer
- Allows the computer to keep track of the interactions
- Allows testing over thousands instead of tens of examples, including:
  - hand-constructed test suites
  - naturally occurring corpus data

Why corpus data?

- No linguist can anticipate all relevant example types to test.
- English Resource Grammar (Flickinger 2000) encoded the expectation that adjectives can’t be pied-piped in free relatives.
- Baldwin et al (2005) found this example by processing a sample of the BNC with the ERG:
  - @However pissed off we might get from time to time, though, we’re going to have to accept that Wilko is at Elland Rd. to stay.

Multiple frameworks

- HPSG: LKB (Copestake 2002), TRALE (Meurers et al 2002)
- LFG: XLE (Maxwell and Kaplan 1996)
- CCG: OpenCCG (Baldridge and Kruijff 2003)
- MP: Minimalist Grammar (Stabler 2000; cf Churng 2006)
- ...

Requirements

- Stable formalism
  - Distinguish formalism from theory
- Parsing, generation, and grammar development tools
- Test suite management tools

Incremental development

- Have to start somewhere
- Selection of where to go next can be
  - theory driven (test suites mostly hand constructed)
  - application driven (test suites combine constructed and naturally occurring data)
- Inertia: Once a decision is made, exploring other options requires a big commitment

Enter the Matrix

Bender, Flickinger & Oepen 2002
Flickinger & Bender 2003
Bender & Flickinger 2005
Drellishak & Bender 2005

Enter the Matrix

- Original motivation was application oriented:
  - We (DELPH-IN) have big grammars for English, Japanese, German
  - Each grammar combines information which looks language-specific with information that looks more general
  - Can we reuse the general parts of existing grammars to reduce the cost of starting a new one?

Original Matrix

- Early versions of the Matrix focussed on ‘universals’
- Most elaboration on the syntax-semantics interface
- And it helped! Broad-coverage grammars for Norwegian (Hellan and Haugereid 2003) and Modern Greek (Kordoni and Neu 2005), started from the Matrix, are still growing

But wait, there’s more

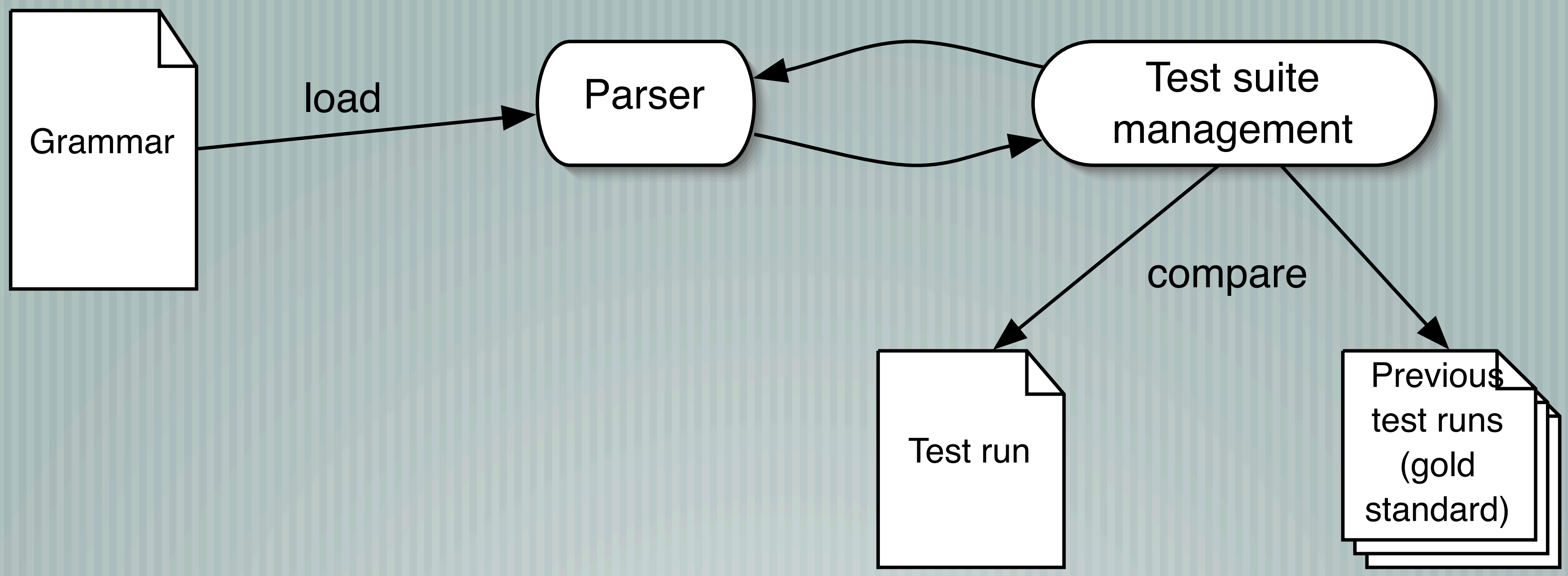
- Many non-universal aspects of language nonetheless recur in many languages
- It’s a shame not to be able to share some code, just because not all languages need it
- Can we apply the same analysis to, e.g., SOV word order everywhere we see it?
- ... crosslinguistic hypothesis testing

Using the Matrix

Division of labor

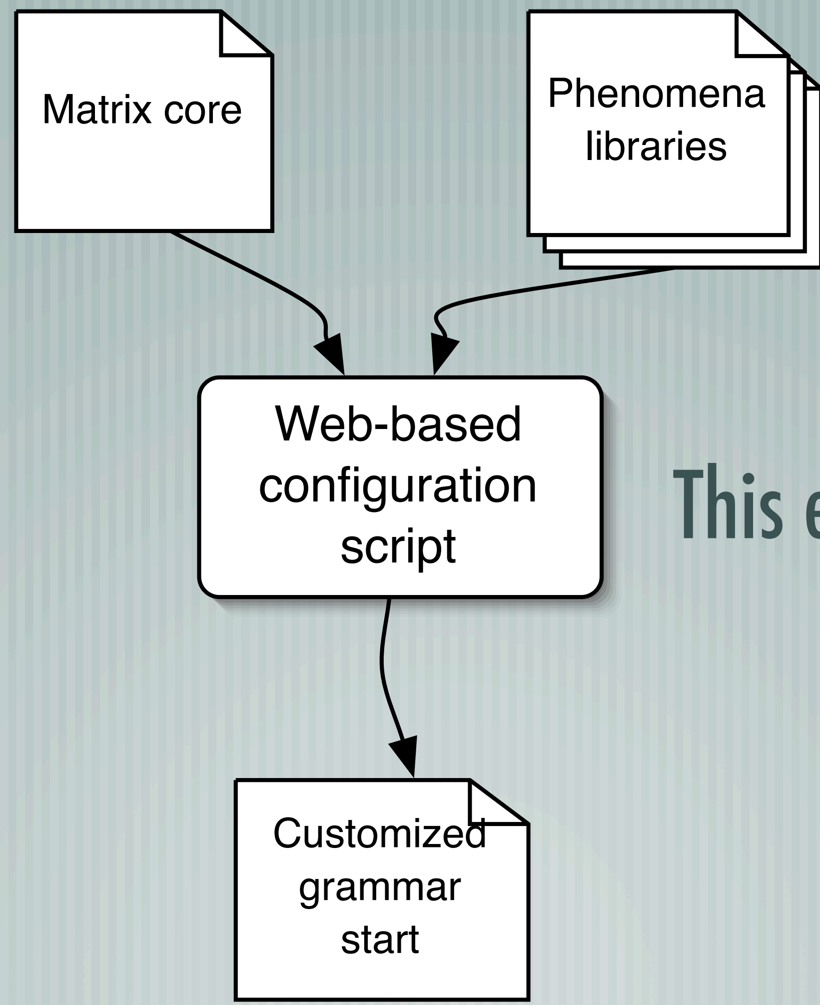
- Declarative grammar (competence): Description of linguistic knowledge
- Parser, generator (performance): Algorithms which use a grammar to analyze or realize strings
- Grammar development tools: GUI tools for visualizing and debugging grammar (LKB: Copestake 2002)
- Test suite management software: Batch process test suite items and analyze results ([incr tsdb()]: Oepen 2001)

Division of labor



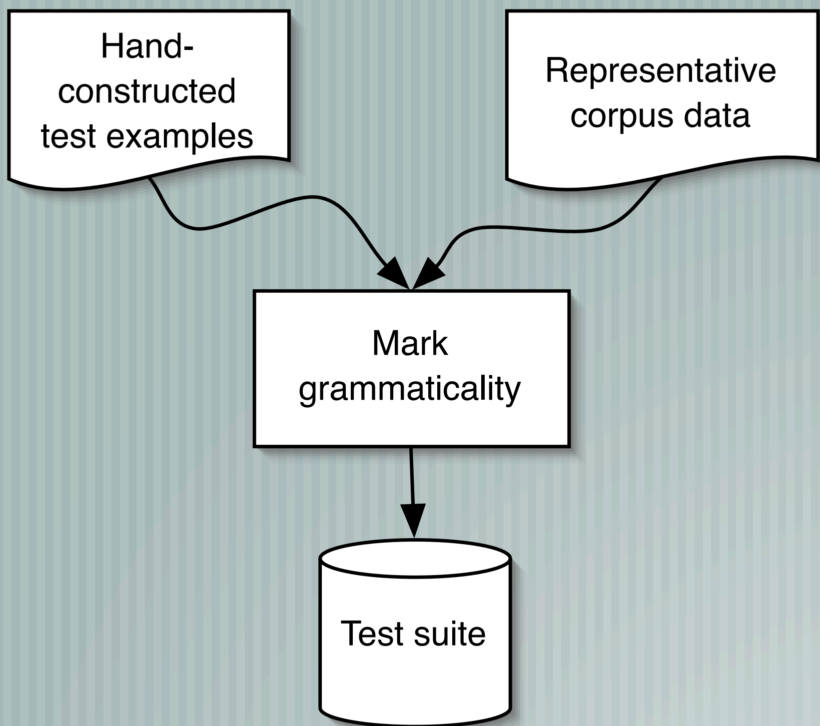
... at a rate of 1000s of sentences per minute!

Matrix as starter-kit

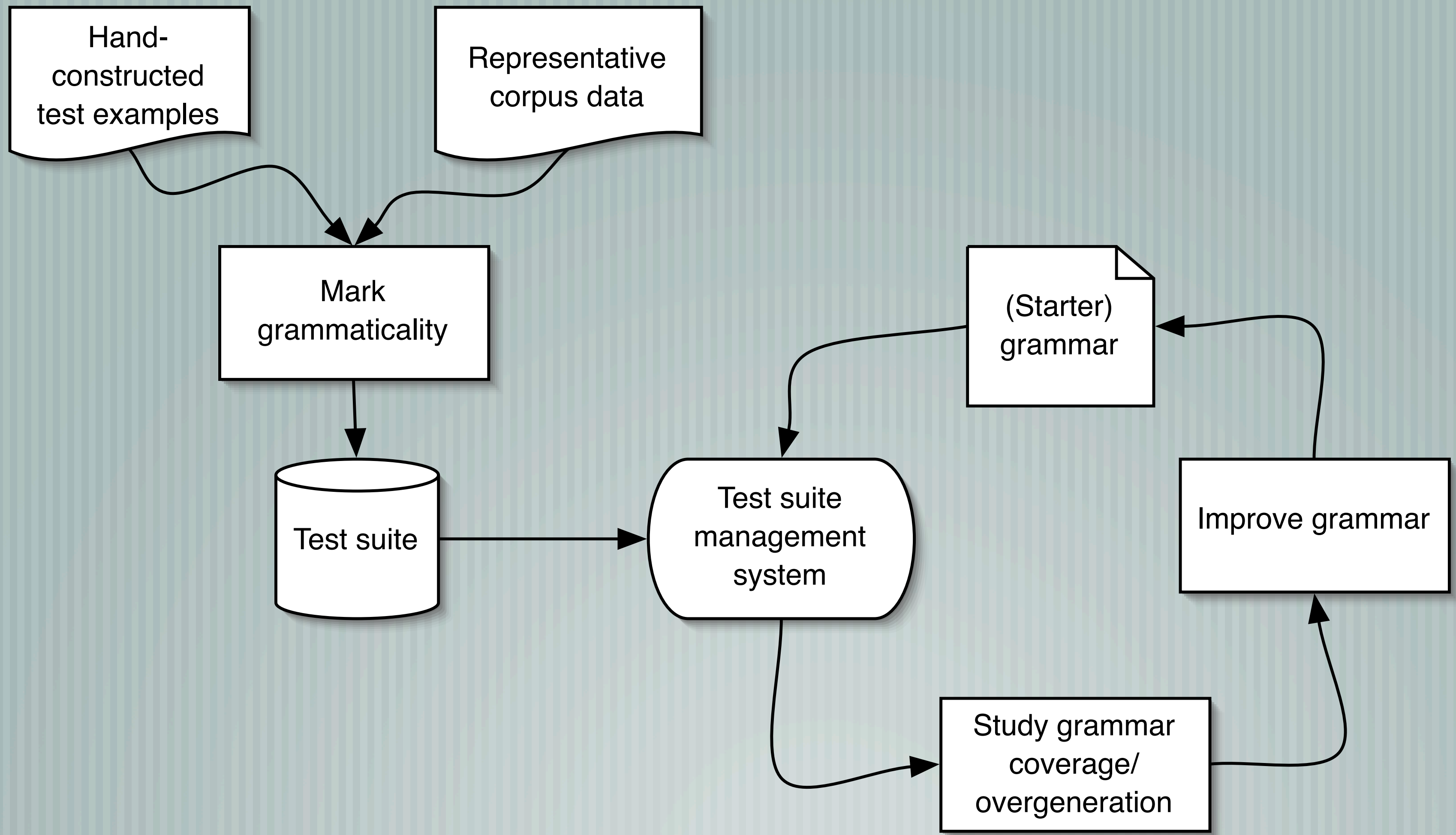


This exists!

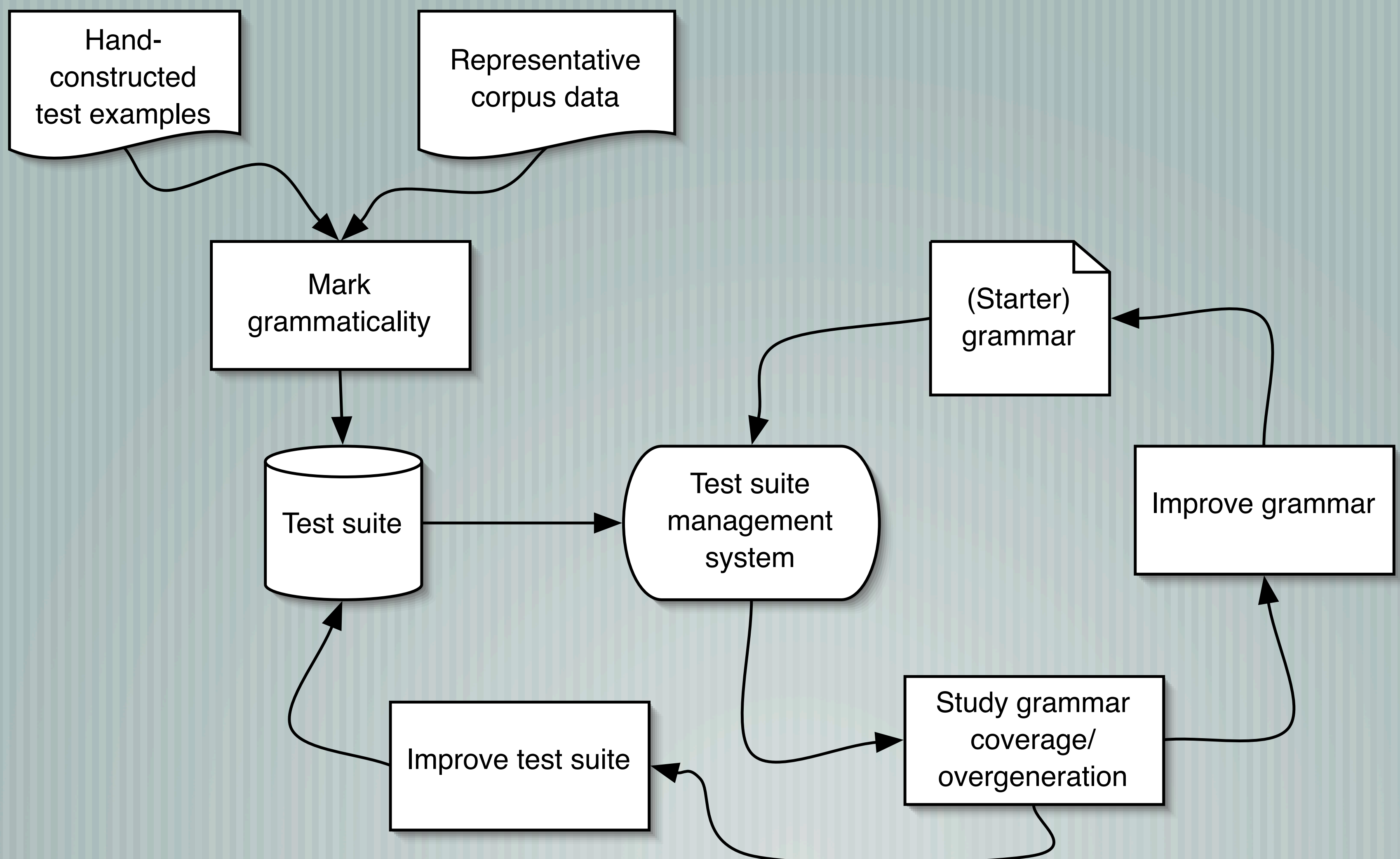
Matrix as starter kit



Matrix as starter kit



Matrix as starter kit



Assumptions

- Have to make some assumptions to get off the ground
- Since the model as a whole is being tested, can only really test hypotheses relative to assumptions
  - This is true of syntax in general, to the extent that we test models by testing their predictions of grammaticality

Assumptions: HPSG

- Monostratal (WYSIWYG) theory; SLASH-passing for long-distance dependencies
- No empty elements
- Rich collection of constructions, with types expressing generalizations across the constructions
- Compositionality: Each constituent gets a semantic representation
- Typed feature-structure formalism

Assumptions HPSG

- X-bar theory: Most phrases are headed, heads select for complements, subjects, and specifiers
- Modifiers select for heads
- Specifiers reciprocally select heads
- ‘Category’ of mother is determined by HEAD value of head daughter and remaining valence requirements
- ...

Assumptions: tdl (LKB)

- No relational constraints: The value of a feature cannot be some function of the value of another (other than equality)
- Any given phrase structure rule has fixed arity.
- Monotonic compositionality: No semantic information lost
- Tectogrammatic/phenogrammatic equivalence: The yield of the tree gives the surface string in order
- ...

Assumptions: Matrix

- Binary branching
- All nouns have associated quantifiers (overt or covert)
- All languages distinguish subjects from other verbal arguments
- All languages have some form of ‘intonation questions’
- ...

Barking up the wrong tree?

- We almost certainly are, at least in some respects
  - It would surprising to be right about so many things
- So why put in all the effort?
  - Test suites are reusable resources
  - Learn things about languages, even if the model eventually fails
  - When it fails, learn about why

Crosslinguistic hypotheses

- The Matrix core contains constraints expected to be useful across all languages
  - Semantic compositionality
  - Valence patterns
  - Superset of part of speech types
  - ...

Typological ‘libraries’

- The libraries contain sets of alternate realizations of specific phenomena
  - Word order
  - Negation
  - Yes-no questions
  - Coordination

Word order

- Major constituent order
- If determiners are present, Det-Noun order
- If adpositions are present, P-NP order
- If auxiliaries are present, aux-V order
- If question particles are present, Q-S order

Yes-no questions

- Matrix-clause only (for now)
  - Subject verb inversion
  - Question particles
  - Intonation only

Sentential negation

- Negative adverbs (independent or selected)
- Negative affix (main or auxiliary verbs)
- If both: always both, complementary distribution, always adverb, always inflection, optionally either

Coordination

- Number of marks
- Position of marks
- Type of marks
- Categories that can be coordinated with that strategy

Crosslinguistic hypotheses

- Aim to handle all known variants on each phenomenon
- Aim for cross-compatibility of the libraries
- Explore where cross-compatibility fails
- Harmonize semantic representations

Isn’t that a lot of grammars?

- Hundreds of thousands, just with the libraries implemented so far, as against 6,000 languages currently spoken today
- Note that there are more than 6,000 possible human languages
- Still, most of our grammars have to be highly unlikely
- We hope this approach will provide an interesting arena in which to explore typological tendencies and universals

Do libraries = parameters?

- At a high enough level of abstraction, yes.
- But:
  - Our libraries handle one phenomenon at a time
  - Necessitated by commitment to handling idiosyncrasies and broad generalizations in one coherent grammar

The other modularity question

- Our libraries correspond to phenomena it makes sense to ask a linguist about
- Adding a library generally involves modifying existing libraries

Example: Word order

- SOV order: comp-head rule
- SOV order plus prepositions: comp-head rule, PP rule
- SOV order plus prepositions plus sentence-initial question particles: comp-head rule, PP | CP rule
- SOV order, prepositions, sentence-initial question particles, pre-verbal auxiliaries: comp-head rule, PP | CP | AuxV rule

Example: Negation

- Adding the negation library turned up a bug in the question library
  - *The cat did didn't chase the dog
- "didn't" in the string above is the output of two lexical rules, one for the -n't suffix and one which adds question semantics
- "did" is seleting for "not" as its first complement
- the question rule lost the information that "didn't" isn't "not"

The other modularity question

- Our libraries correspond to phenomena it makes sense to ask a linguist about
- Adding a library generally involves modifying existing libraries
- Why?
  - *un système où tout se tient*
  - HPSG architecture
- Perhaps we'll be able to refactor when we're done

Evaluation

- How can you tell if it works?
  - Build lots of grammars, test against real data, see where the Matrix-provided constraints are revised or ignored (Ling 567)
  - But first: Create a resource of abstract strings annotated with grammaticality predictions per language type to test interaction of existing libraries. (Poulson 2006)

Conclusion

- Grammar engineering draws on theoretical results in syntax
  - Initial motivation of frameworks to try
  - Data of interest
  - Proposals of analyses
- Theoretical syntax can turn to grammar engineering for large-scale validation of ideas