

# The Matrix: Future Directions

## Wrap up

---

Ling 567

June 3, 2013

# Overview

---

- Wrap up/reflections
- Matrix: Future directions
  - More libraries
  - More robust MMT
  - Applications, including language documentation

# Goals: Of Grammar Engineering

---

- Build useful, usable resources
- Test linguistic hypotheses
- Represent grammaticality/minimize ambiguity
- Build modular systems: maintenance, reuse

# Goals: Of this course

---

- Mastery of tfs formalism
- Hands-on experience with grammar engineering
- A different perspective on natural language syntax
- Practice building (and debugging!) extensible system
- Contribute to on-going research in multilingual grammar engineering

# Reflections

---

- Where have the analyses provided by the Matrix (or suggested by the labs) seemed like a good fit?
- Where have they been awkward?
- What have you learned in this class about syntax?
- ... about knowledge engineering for NLP?
- ... about computational linguistics in general?
- ... about linguistics in general?
- What did working with a test corpus show you about the process of scaling to real-world text?

# Feedback: Pair projects

---

- How did you divide the work?
- In what ways was having a partner helpful?
- Would you have learned more working on your own?

# Future directions overview

---

- More libraries (and semantic harmonization)
- How this class will evolve
- MT: Auto-generated transfer rules, typological seeding of statistical NLP (including SMT)
- Lexical acquisition
- Ontological annotation
- Matrix-ODIN Mash-up

# More libraries

---

- New this year: Information structure, multiple inheritance of lexical types, updates to negation
- Next up?
  - Demonstratives
  - Extensions/retrofits to questions, coordination
  - (more) extensions to word order
  - Non-verbal predicates
  - Intersective modifiers
  - Numeral classifiers
  - More verb subcategorization
  - Embedded clauses



# How to make a library

---

1. Delineate a phenomenon
2. Survey the typological literature: How is this phenomenon expressed across the world's languages?
3. Review the syntactic literature for analyses of the phenomenon in its various guises
4. Design target semantic representations
5. Develop HPSG analyses for each variant and implement in tdl
6. Decide what information is required from the user to select the right analysis, and extend questionnaire accordingly
7. Extend customization script to add tdl based on questionnaire answers
8. Add regression tests documenting functionality
9. Add prose documenting how to use

# How to evaluate a library

---

- Pseudo-languages
- Illustrative languages
- Held-out languages
- Test suites
- Choices files
- Error analysis

# More libraries/reflection from current class

---

- What do you most wish was available in the customization system, based on what came up in your test suite?
- In your test corpus?

# Evolution of 567

---

- New phenomena: Wh-questions, relative clauses, while-clauses ...?
- Ever bigger jump start --- reaching the limit on this one?
  - Would working in groups of three make it possible to get to even bigger grammar fragments?
- Relatively new/how did these work out?:
  - Partnership with field linguists
  - Work with small corpora
- Coverage-driven labs seem most satisfying (MT demo, corpus coverage). Is this true? Can the course be rebalanced to do more of this?

# Lexical acquisition

---

- How can we import lexical entries from other linguistic resources (e.g., FIELD lexicons, ODIN)?
- How big do the grammars have to get before we can embark on (semi-)automated lexical acquisition?
- To what extent do the lexical properties of translational equivalents predict lexical properties in another language?
- How can we most effectively leverage human effort?
- How do we know when we're missing an appropriate type?

# Autogenerated transfer rules

---

- Identify “grammaticalized” differences in MRSs
- “Publish” choices along these dimensions for each grammar
- Create a library of transfer rules from property to property:
  - pro-drop to pronouns (and vice versa)
  - mismatches in demonstrative distinctions
  - can <> the possibility exists
  - hurt/cause feel+pain/cause harm

# Autogenerated transfer rules

---

- Use language-specific pred values
- Create transfer rules on the basis of PanDictionary or other lexical resources
- Measure the extent of translation divergence (Francesca Gola's MS thesis work)
- Use bitexts and statistical methods to detect word pairs requiring more than straight pred-mapping transfer rules

# Ontological annotation

---

- Annotate grammars with links to GOLD (Farrar & Langendoen 2003)
  - Locate which constraints contribute to which phenomena
  - Index analyses for discovery in grammars and treebanks
- Annotations in Matrix core
- Annotations in customization system
- Support for user annotation



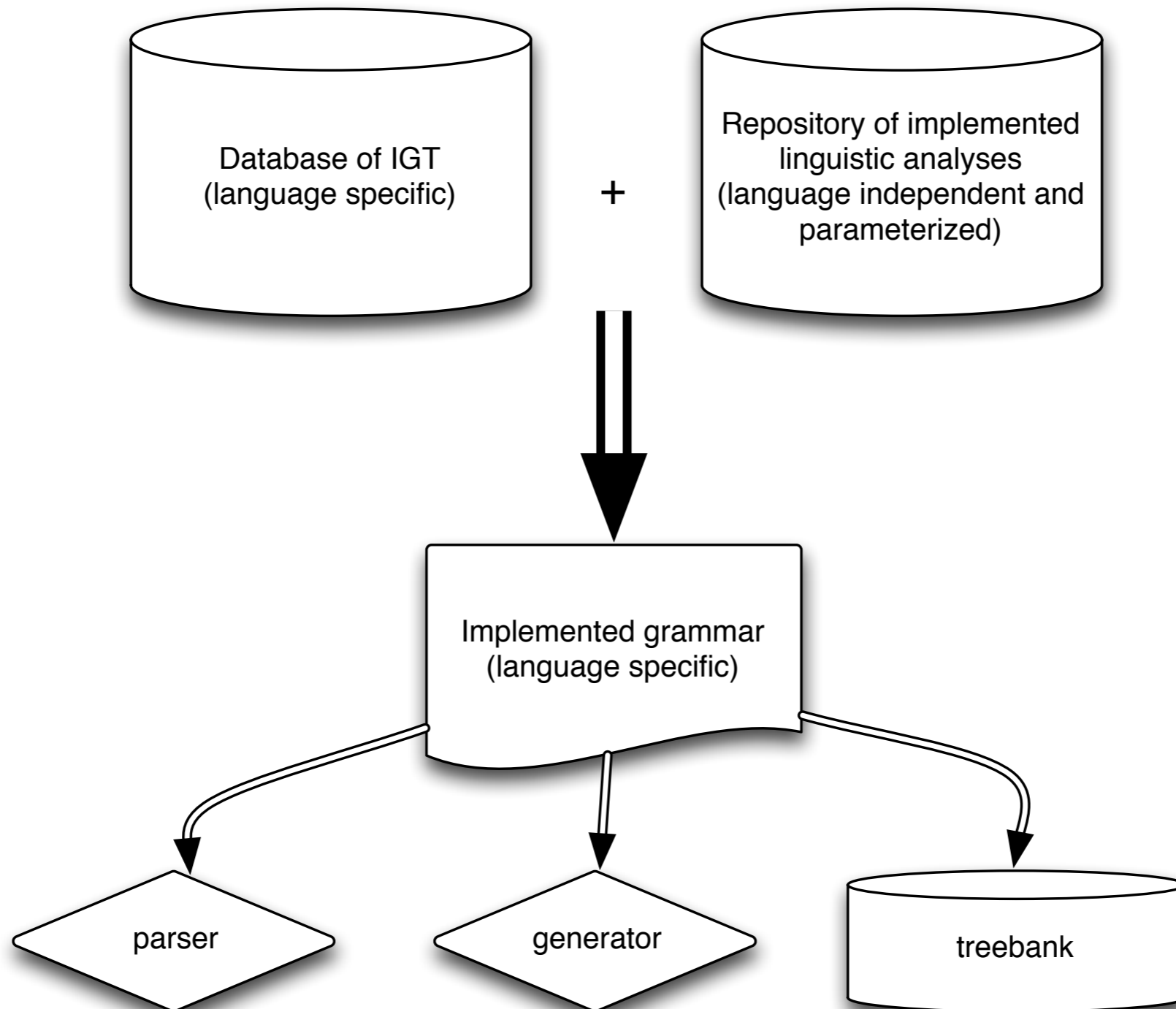
# AGGREGATION: Research goals

---

- Precision implemented grammars are a kind of structured annotation over linguistic data (cf. Good 2004, Bender et al 2012).
- They map surface strings to semantic representations and vice-versa.
- They can be used in the development of *grammar checkers* and *treebanks*, making them useful for language documentation and revitalization (Bender et al 2012)
- But they are expensive to build.
- The AGGREGATION project asks whether existing products of documentary linguistic research (IGT collections) can be used to boot-strap the development of precision implemented grammars.

# Combining linguistic knowledge

---



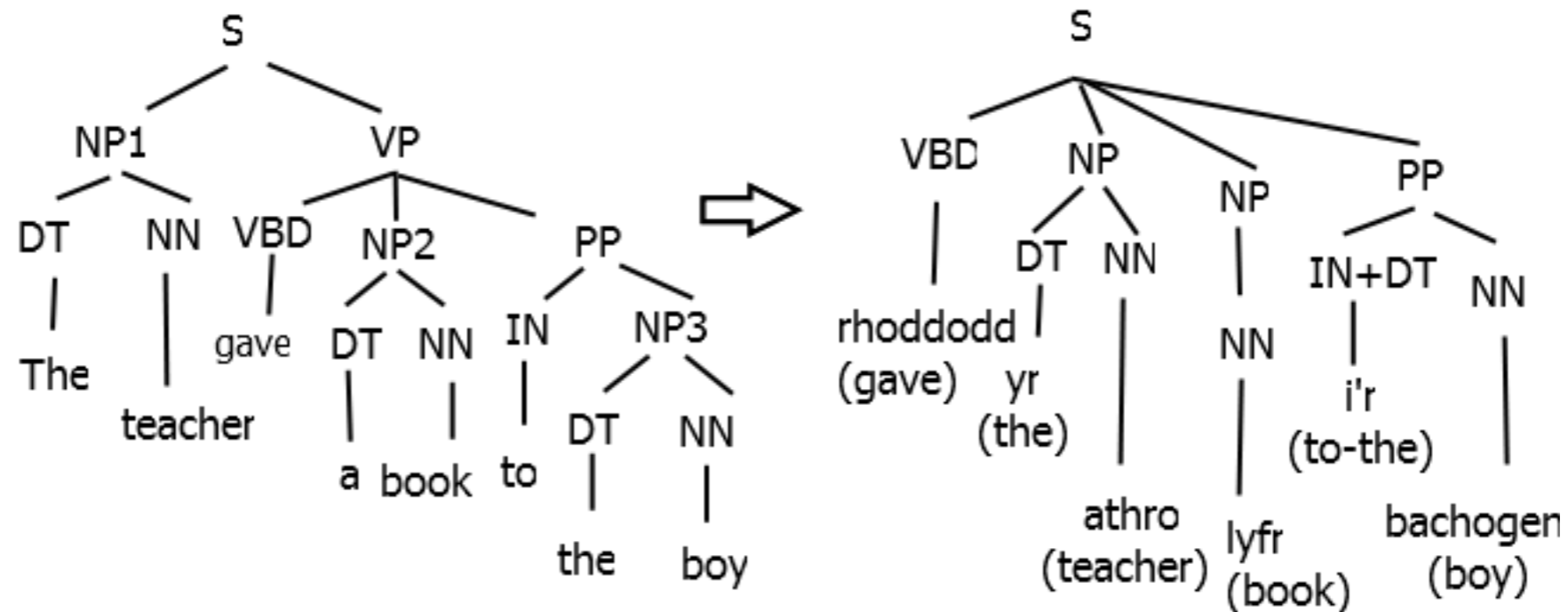
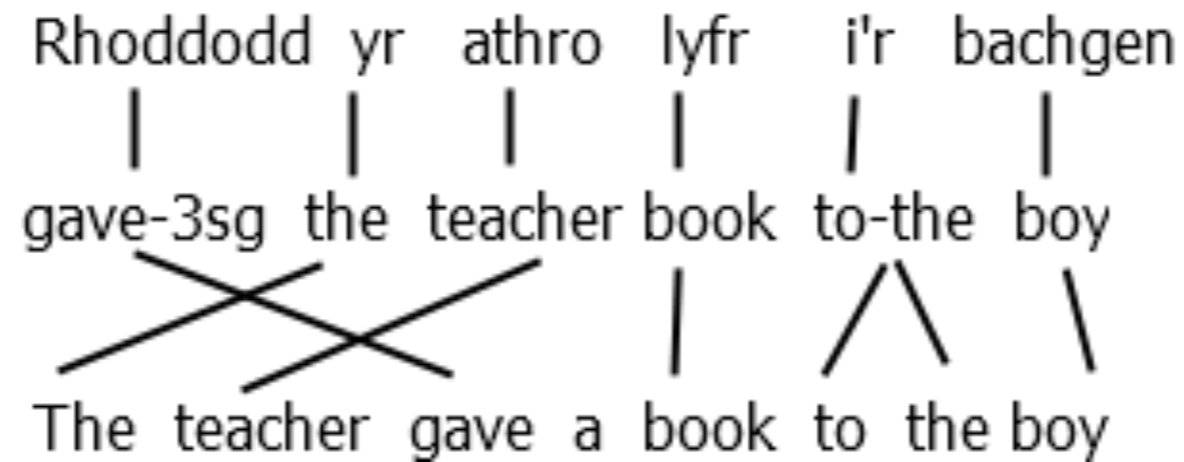
# RiPLEs: Goals

---

- RiPLEs: information engineering and synthesis for Resource Poor Languages
- Support rapid development of NLP resources for RPLs by bootstrapping through IGT
- Support cross-linguistic study through creating ‘language profiles’ based on IGT analysis

(Xia & Lewis 2007, Lewis & Xia 2008)

# RiPLEs: IGT projection methodology



(Xia & Lewis 2009)

# RiPLEs: Results

---

Table 3: Experiment 1 Results (Accuracy)

	WOrder	VP +OBJ	DT +N	Dem +N	JJ +N	PRP\$ +N	Poss +N	P +NP	N +num	N +case	V +TA	Def	Indef	Avg
basic CFG	0.8	0.5	0.8	0.8	1.0	0.8	0.6	0.9	0.7	0.8	0.8	1.0	0.9	0.800
sum(CFG)	0.8	0.5	0.8	0.8	0.9	0.7	0.6	0.8	0.6	0.8	0.7	1.0	0.9	0.762
CFG w/ func	0.9	0.6	0.8	0.9	1.0	0.8	0.7	0.9	0.7	0.8	0.8	1.0	0.9	0.831
both	0.9	0.6	0.8	0.8	0.9	0.7	0.5	0.8	0.6	0.8	0.7	1.0	0.9	0.769

Table 5: Word Order Accuracy for 97 languages

# of IGT instances	Average Accuracy
100+	100%
40-99	99%
10-39	79%
5-9	65%
3-4	44%
1-2	14%

(Lewis & Xia 2008)

# Word order options

---

- Lewis & Xia 2008, Dryer 2011 (WALS)
  - SOV
  - SVO
  - OSV
  - OVS
  - VSO
  - VOS
  - no dominant order

- Grammar Matrix
  - SOV
  - SVO
  - OSV
  - OVS
  - VSO
  - VOS
  - Free (pragmatically determined)
  - V-final
  - V-initial
  - V2

# Word order in the Grammar Matrix

---

- More than a simple descriptive statement
- Affects phrase structure rules output by the system, but also interacts with other libraries (e.g., argument optionality)
- These phrase structure rules help model the mapping of syntactic to semantic arguments
- Underlying word order is not reflected in every sentence; testsuites won't have the same distribution as naturally occurring corpora
- Matrix users advised to choose fixed word order if deviations from that order can be attributed to specific syntactic constructions

# Methodology

---

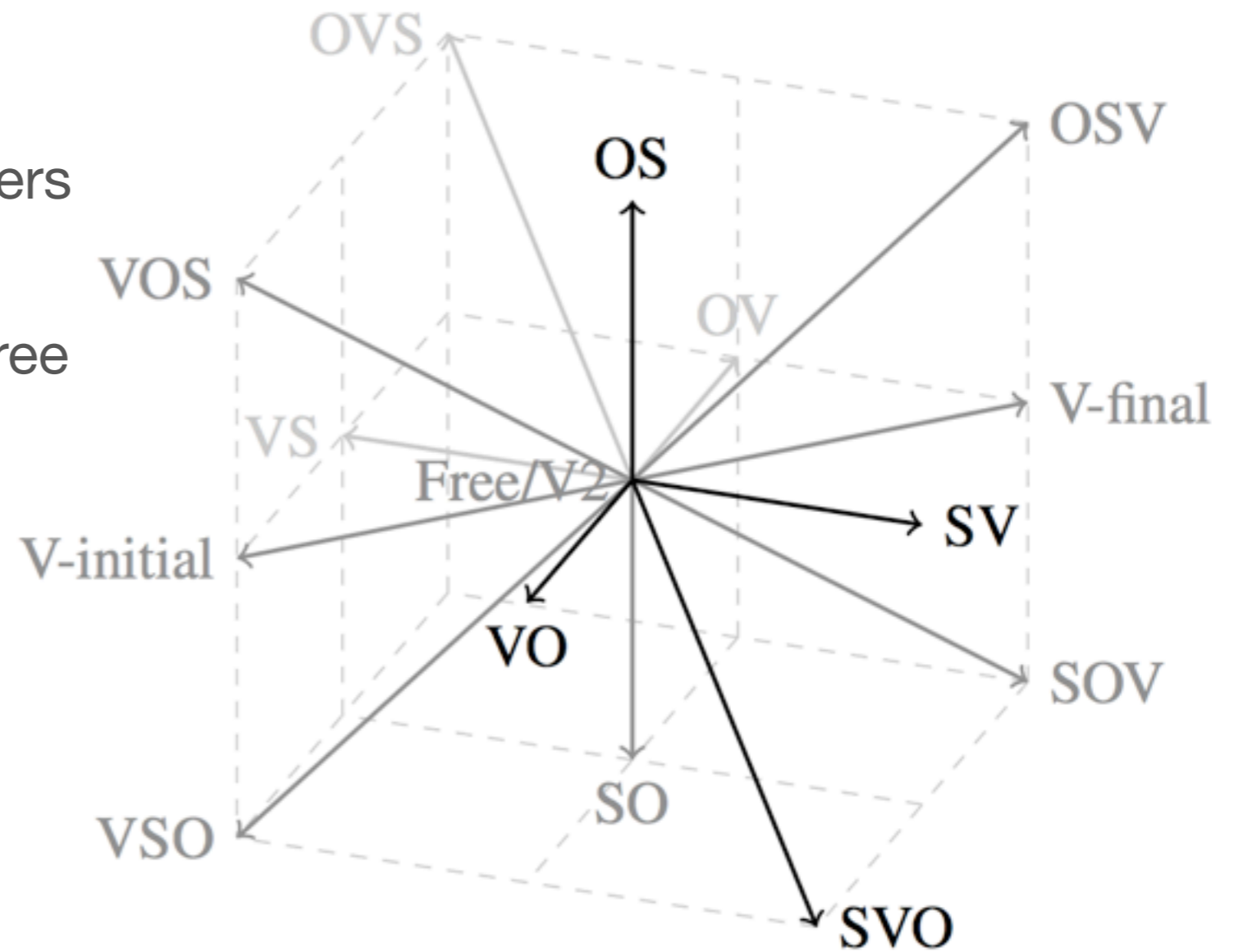
- Parse English translation and project the parsed structure onto the language line (per RiPLEs)
- Add -SBJ and -OBJ function tags to the English parse trees (by heuristic), and project these too
- *Observed word orders:* counts of the 10 patterns SOV, SVO, OSV, OVS, VSO, VOS, SV, VS, OV, and VO in the source language trees
- Decompose SOV, SVO, OSV, OVS, VSO, VOS into order of S/O, S/V and O/V



# Methodology

---

- SOV, SVO, OSV, OVS, VSO, VOS
- Measure Euclidean distance to positions of canonical word orders
- In a separate step, distinguish free from V2



# Dev and test data

---

- 31 testsuite + choices file pairs, developed in Linguistics 567 at UW (Bender 2007)

	DEV1	DEV2	TEST
Languages	10	10	11
Grammatical examples	16–359 (median: 91)	11–229 (median: 87)	48–216 (median: 76)
Language families	Indo-European (4), Niger-Congo (2), Afro-Asiatic, Japanese, Nadahup, Sino-Tibetan	Indo-European (3), Dravidian (2), Algic, Creole, Niger-Congo, Quechuan, Salishan	Indo-European (2), Afro-Austro-Asiatic, Austronesian, Arauan, Carib, Karvelian, N. Caucasian, Tai-Kadai, I

# Results

---

- Compare to most-frequent-type (SOV, Dryer 2011)

Dataset	Inferred WO	Baseline
DEV1	0.900	0.200
DEV2	0.500	0.100
TEST	0.727	0.091

- Sources of error:
  - Testsuite bias
  - Misalignment in projections

# Case system options in the Grammar Matrix: Case marking on core arguments of (in)transitives

---

- None
- Nominative-accusative
- Ergative-absolutive
- Tripartite
- Split-S
- Fluid-S
- Split conditioned on features of the arguments
- Split conditions on features of the V
- Focus-case (Austronesian-style)
- The choice among these options makes further features available on the lexicon page, including case frames
- There is always the option to define more cases and case frames

# Two methods

---

- GRAM: Assume Leipzig Glossing Rules-compliance (Bickel et al 2008)
- Search gloss line for case grams, and assign system as follows:

Case system	Case grams present					
	NOM	V	ACC	ERG	V	ABS
none						
nom-acc		✓				
erg-abs					✓	
split-erg (conditioned on V)		✓			✓	

- SAO: Use RiPLEs to identify S, A, and O arguments
- Collect most frequent gram for each
- Compare most frequent grams across S/A/O to determine case system

# Results

---

Dataset	GRAM	SAO	Baseline
DEV1	0.900	0.700	0.400
DEV2	0.900	0.500	0.500
TEST	0.545	0.545	0.455

- GRAM confused by non-NOM/ACC style glossing
- SAO confused by testsuite bias (spurious most-frequent elements)
- SAO confused by alignment errors (e.g. case marking adpositions)

# Summary

---

- First steps towards our long-term goal: Automatically create working grammar fragments from IGT, by taking advantage of
  - Grammar Matrix customization system's mapping of relatively simple language description files to working grammars
  - Linguistic analysis encoded in IGT
  - RiPLes methodology for further enriching IGT
- Resulting grammars are of interest for testing the Grammar Matrix as a set of typological hypotheses
- And potentially for field grammarians (when built-out) as they can support the creation of treebanks and exploration of corpora for unanalyzed phenomena

# Overview

---

- Wrap up/reflections
- Matrix: Future directions
  - More libraries
  - More robust MMT
  - Applications, including language documentation