Making Computers Help Linguists: Grammar Engineering for Linguistic Hypothesis Testing, Linguistic Typology, and Language Documentation

Emily M. Bender
University of Washington

Arizona Linguistics Circle 12
October 14, 2018

UNIVERSITY of WASHINGTON
DEPARTMENT OF LINGUISTICS

UWNLP

DELPH-IN

Acknowledgments

- Grammar Matrix collaborators: Dan Flickinger, Stephan Oepen, Scott Drellishak, Laurie Poulson, Kelly O'Hara, Michael Goodman, Antske Fokkens, Joshua Hou, Safiyyah Saleem, Daniel Mills, Sanghoun Song, Joshua Crowgey, Scott Halgrim, David Wax, Varya Gracheva, Laurie Dermer, Michael Haeger, Olga Zamaraeva, Kristen Howell, Elizabeth Nielsen

- AGGREGATION collaborators: Fei Xia, Michael Goodman, Joshua Crowgey, Olga Zamaraeva, Ryan Georgi, Kristen Howell, Swetha Ramaswamy, Haley Lepp

- Students in Ling 567 (since 2004) and 575 (2015)

- NSF grants BCS-0644097, BCS-1160274, BCS-1561833

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Main claims

- Grammar engineering allows us to off-load the tedious part of verifying analyses to a computer
- The Grammar Matrix customization system speeds up the process of grammar engineering
  - … while also providing a testbed for typological generalizations
- Grammar engineering can be useful in work with endangered and other understudied languages
- We can automate the first steps of grammar development by inferring answers to the Grammar Matrix questionnaire from IGT
  - … and this process itself provides useful insight into data collections

Overview

- Grammar engineering
- The LinGO Grammar Matrix
- AGGREGATION

Grammar Engineering

- The development of grammars-in-software: morphology, syntax, semantics
- “Precision grammars”
    - Encode linguistic analyses
    - Human- and machine-readable
    - Model grammaticality
    - Map strings to underlying representations
    - Can be used for both *parsing* and *generation*

Grammar Engineering: Frameworks

- Precision grammars have been built by/in/with
  - HPSG in ALE/Controll (Götz & Meurers 1997; CoreGram: Müller 2015)
  - LFG (ParGram: Butt et al 2002)
  - F/XTAG (Doran et al 1994)
  - SFG (Bateman 1997)
  - GF (Ranta 2007)
  - OpenCCG (Baldridge et al 2007)
  - Proprietary formalisms and Microsoft and Boeing and IBM
- On implementation of MP, see e.g. Stabler 2001, Fong 2015, Herring 2016

DELPH-IN: Deep Linguistic Processing in HPSG Initiative (www.delph-in.net)

- Informal, international consortium established in 2002
- Shared repository of open-source, interoperable resources
- Framework/formalisms:
  - Head-Driven Phrase Structure Grammar (HPSG; Pollard & Sag 1994)
  - Minimal Recursion Semantics (MRS; Copestake et al 2005)
  - DELPH-IN joint reference formalism (Copestake 2002a)

DELPH-IN: Deep Linguistic Processing in HPSG Initiative (www.delph-in.net)

- **Grammars**: ERG (Flickinger 2000, 2011); Jacy (Siegel, Bender & Bond 2016); SRG (Marimon 2010); gCLIMB (Fokkens 2014); Indra (Moejadi 2018); ...
- **Parsing & Generation**: LKB (Copestake 2002b); PET (Callmeier 2002); ACE (http://sweaglesw.org/linguistics/ace); Agree (Slayden 2012)
- **Regression testing**: [incr tsdb()] (Oepen 2001)
- **Treebanking**: Redwoods (Oepen et al 2004), FFTB (Packard 2015)
- **Applications**: e.g., MT (Oepen et al 2007), QA from structured knowledge sources (Frank et al 2007), Textual entailment (Bergmair 2008), ontology construction (Nichols et al 2006) and grammar checking (Suppes et al 2012), robot control language (Packard 2014), sentiment analysis (Kramer & Gordon 2014), ...

HPSG in one slide

- Key references: Pollard & Sag 1987, Pollard & Sag 1994, Sag, Wasow & Bender 2003 (textbook)
- Phrase structure grammar: Like CFG but with elaborate feature structures instead of atomic node labels
- Monostratal/surface oriented: One structure per input item (no movement), with both syntactic and semantic information
- Lexicalist: Rich information in lexical entries (+ type hierarchy to capture generalizations)
- Core & periphery: Construction inventory includes both very general and very idiosyncratic rules

Minimal Recursion Semantics in one slide

- Key references: Copestake et al 2005, Bender et al 2015
- Underspecified description of logical forms
- Captures predicate-argument structure, partial constraints on quantifier scope, morpho-semantic features
- Computationally tractable, grammar-compatible, and linguistically expressive

English Resource Grammar (Flickinger 2000, 2011)
erg.delph-in.net

DELPH-IN

- Under continuous development since 1993
- Broad-coverage: 85-95% on varied domains: newspaper text, Wikipedia, bio-medial research literature (Flickinger et al 2010, 2012; Adolphs et al 2008)
    - Robust processing techniques enable 100% coverage
- Output: derivation trees paired with meaning representations in the Minimal Recursion Semantics framework---English Resource Semantics (ERS)
    - Emerging documentation at moin.delph-in.net/ErgSemantics

English Resource Grammar
erg.delph-in.net

- 1214 release: 225 syntactic rules, 70 lexical rules, 975 leaf lexical types
- Generalizations captured in a type hierarchy
- Both ‘core’ (high frequency) and ‘peripheral’ constructions

```
head_subj_phrase := basic_head_subj_phrase &
  [ HD-DTR.SYNSEM.LOCAL.CAT.VAL.SUBJ < #synsem >,
    NH-DTR.SYNSEM #synsem ].
```

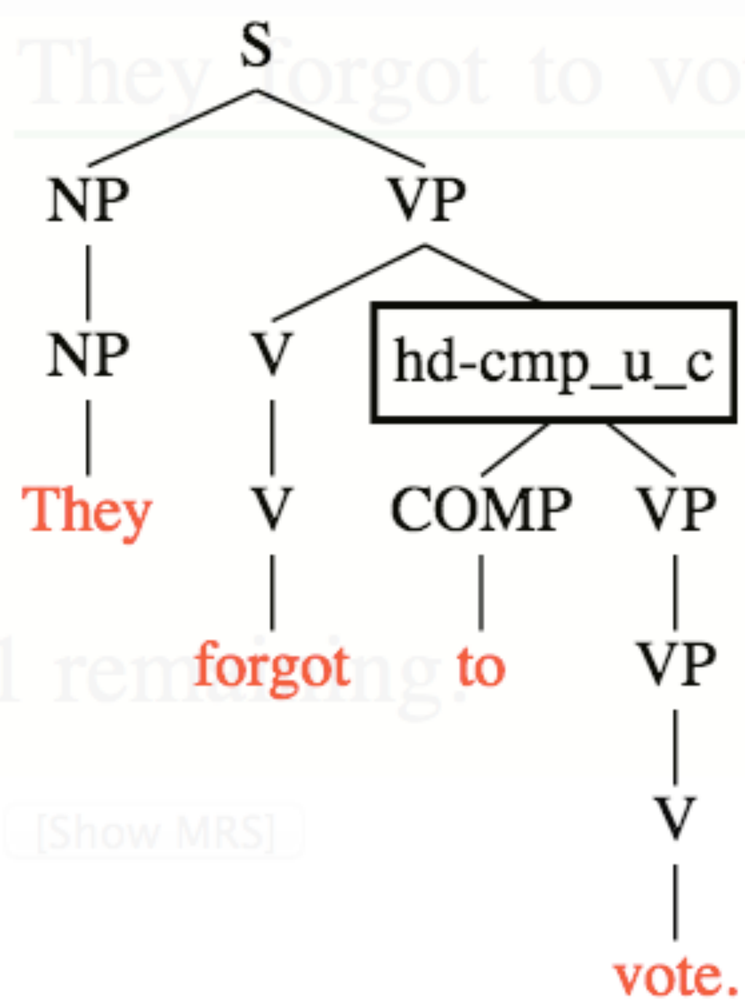
English Resource Grammar
erg.delph-in.net

```
modgap_rel_cl := basic_non_wh_rel_cl &
  [ SYNSEM.LOCAL.CAT.HEAD.MOD < [ LOCAL.CAT.HEAD noun,
                                  --MIN modable_rel,
                                  --SIND #mind ] >,
    ARGS < [ SYNSEM
            [ LOCAL.CONT.HOOK.INDEX.SF prop,
              NONLOC.SLASH 1-dlist &
              [ LIST < mod-local &
                       [ CAT.HEAD mobile & [ MOD < synsem > ],
                         CONT.HOOK [ LTOP #sltop,
                                     INDEX #slind & [ SORT location ],
                                     XARG #xarg ] ] > ] ] ] >,
    ORTH [ FROM #from, TO #to ],
    C-CONT.RELS <! prep_relation &
                  [ LBL #sltop,
                    PRED loc_nonsp_rel,
                    ARG0 #slind & [ E [ TENSE no_tense,
                                        ASPECT no_aspect ] ],
                    ARG1 #xarg & event_or_index,
                    ARG2 #mind & [ SORT basic-entity-or-event ],
                    CFROM #from, CTO #to ] !> ].
```

English Resource Grammar
erg.delph-in.net

```
basic_head_subj_phrase := head_nexus_rel_phrase & head_final_infl & phrasal &
  [ SYNSEM [ LOCAL [ CAT.VAL [ COMPS < >,
                               SPR < >,
                               SUBJ *olist* & < anti_synsem_min >,
                               SPEC #spec,
                               SPCMPS < > ],
                     CONJ cnil ],
             MODIFD.RPERIPH #rperiph,
             PUNCT.PNCTPR #ppair ],
    HD-DTR.SYNSEM [ LOCAL.CAT [ VAL [ COMPS < >,
                                      SPR *olist*,
                                      SPEC #spec ],
                                MC na ],
                    MODIFD.RPERIPH #rperiph,
                    PUNCT [ LPUNCT pair_or_no_punct,
                            PNCTPR #ppair ] ],
    NH-DTR.SYNSEM canonical_synsem &
                [ LOCAL [ CAT [ HEAD subst,
                                VAL [ SUBJ *olist_or_prolist*,
                                      COMPS < >,
                                      SPR *olist* ] ] ],
                  NONLOC [ SLASH 0-dlist,
                           REL 0-dlist ],
                  PUNCT [ LPUNCT pair_or_no_punct,
                          RPUNCT comma_or_rbc_or_pair_or_no_punct,
                          PNCTPR ppair ] ] ].
```

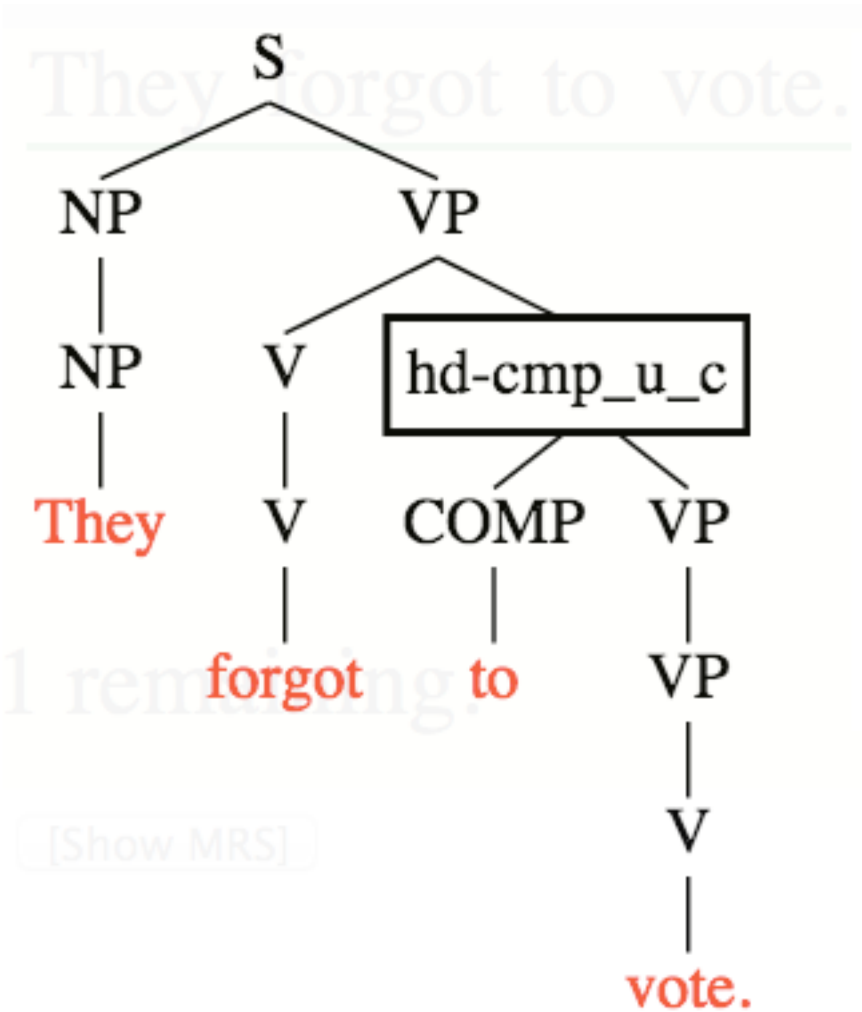
ERG: Examples

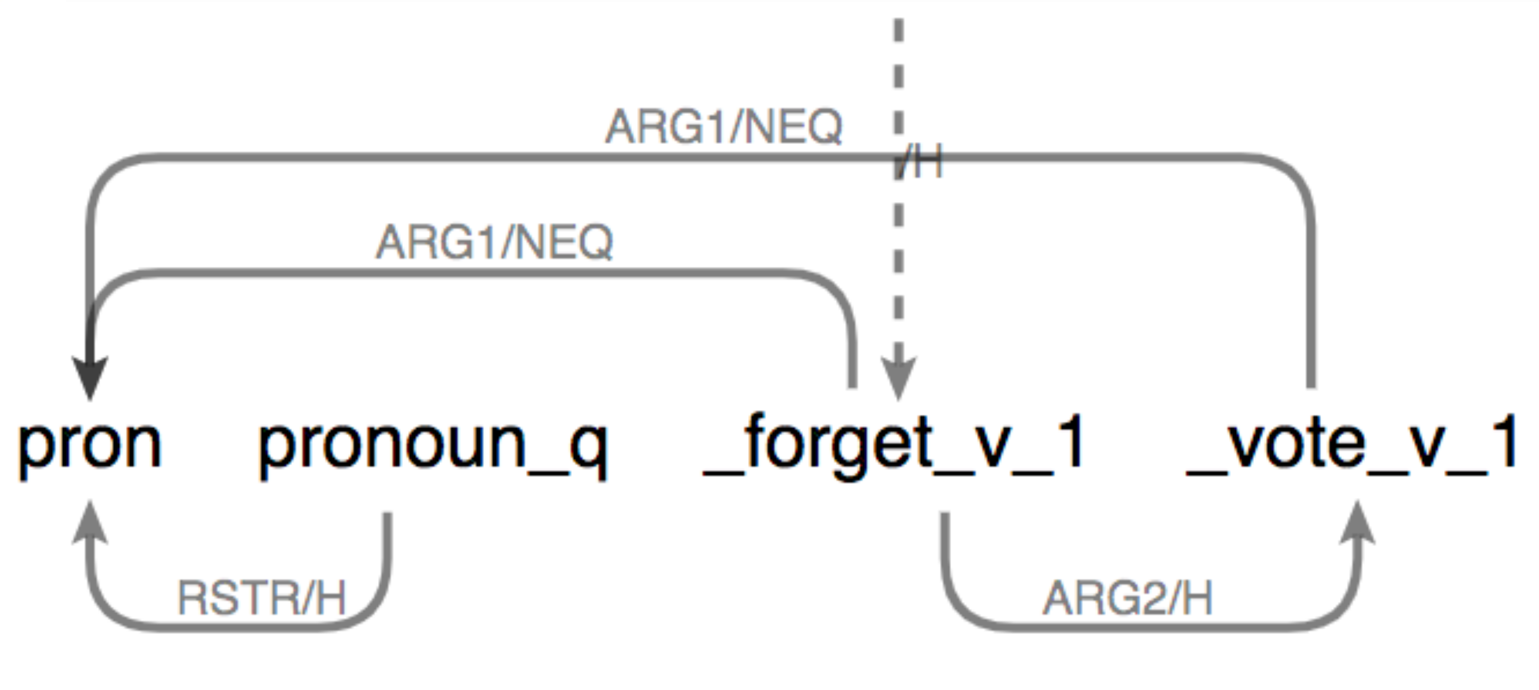


TOP: **h0**
INDEX: **e2**
RELS:
h4:pron_rel(ARG0: **x3**)
h5:pronoun_q_rel(ARG0: **x3**,RSTR: **h6**,BODY: **h7**)
h1:"_forget_v_1_rel"(ARG0: **e2**,ARG1: **x3**,ARG2: **h8**)
h9:"_vote_v_1_rel"(ARG0: **e10**,ARG1: **x3**)

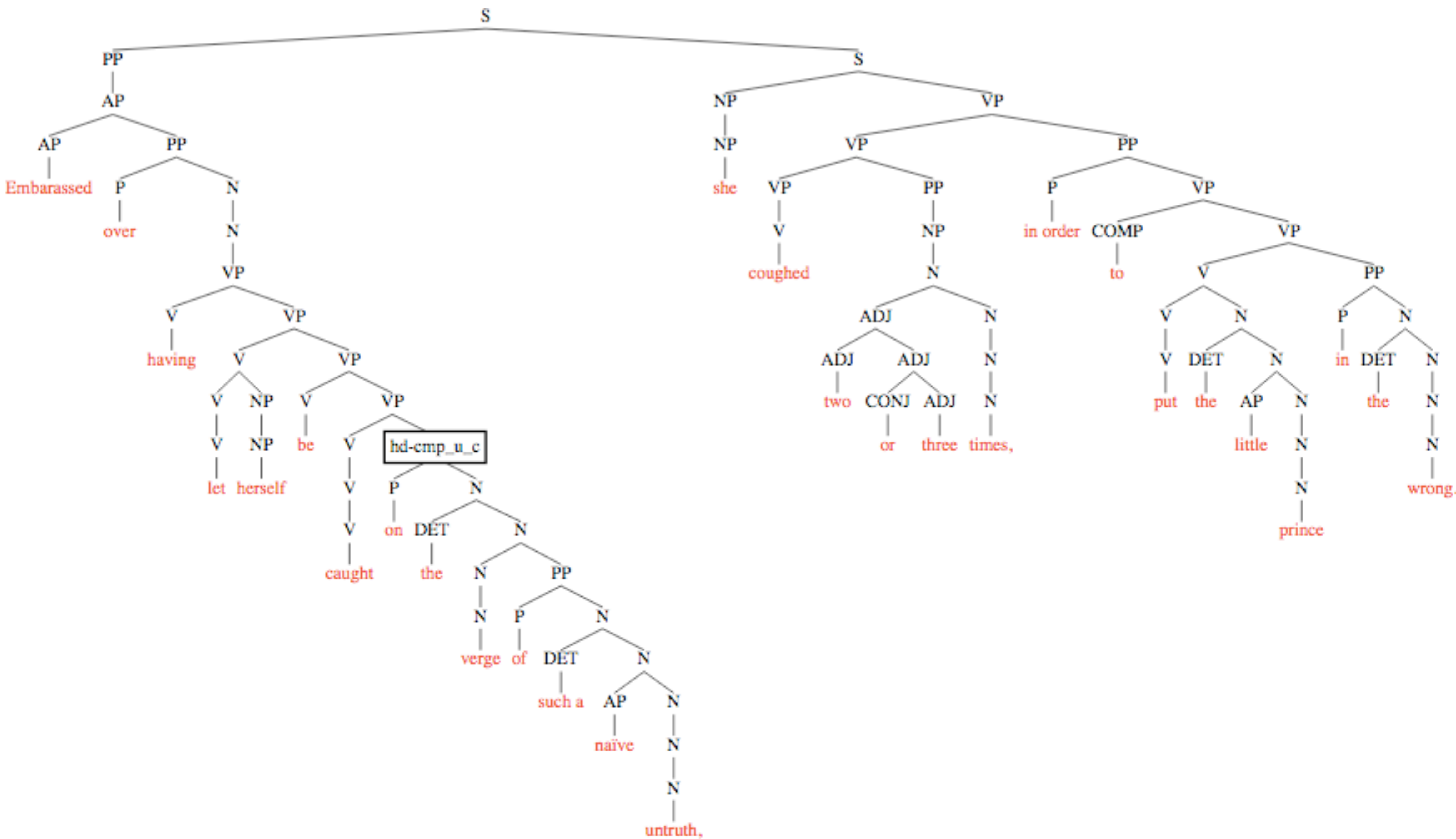
HCONS: **h0** $=_q$ **h1**, **h6** $=_q$ **h4**, **h8** $=_q$ **h9**

ERG: Examples





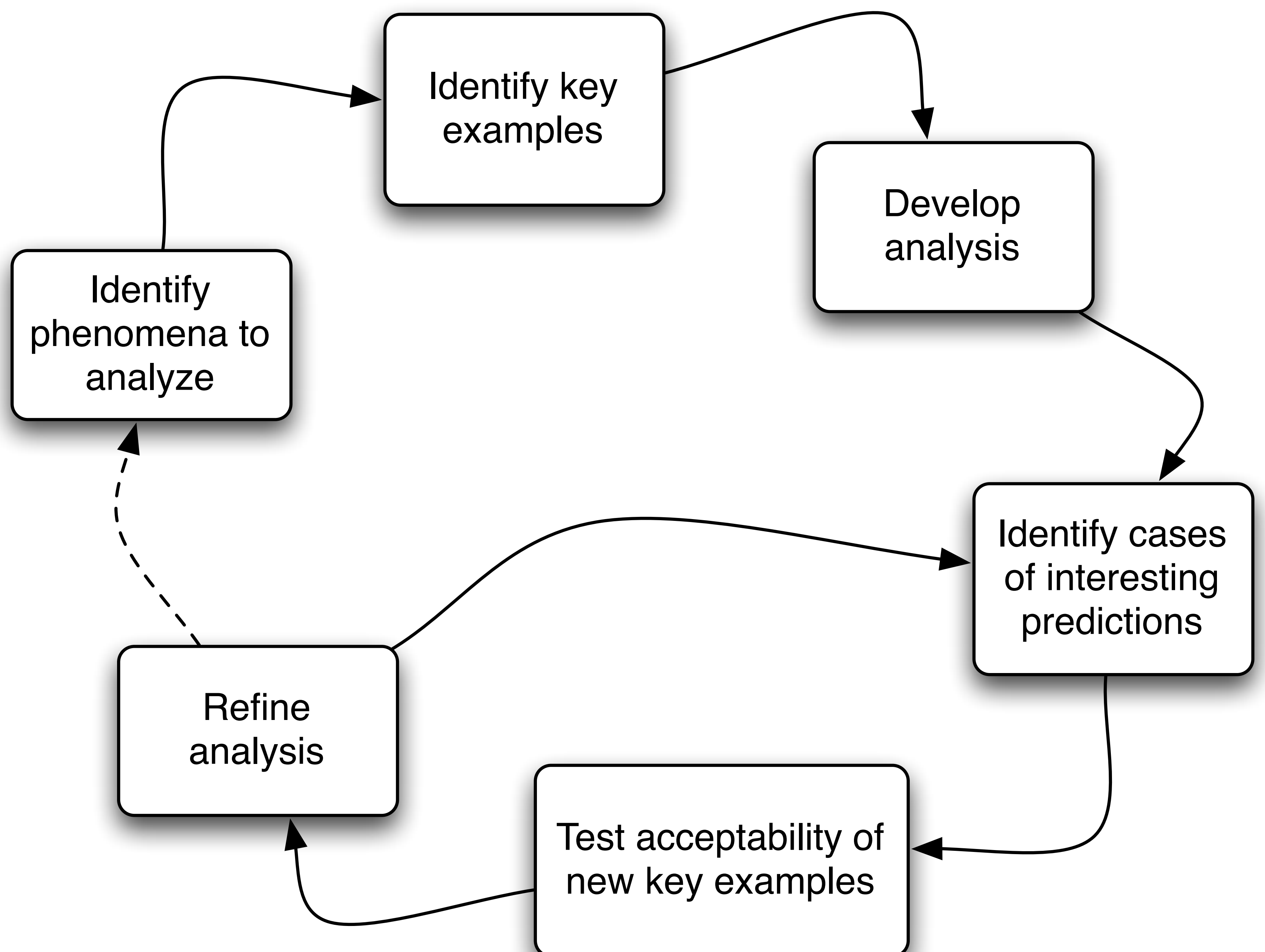
ERG: Examples



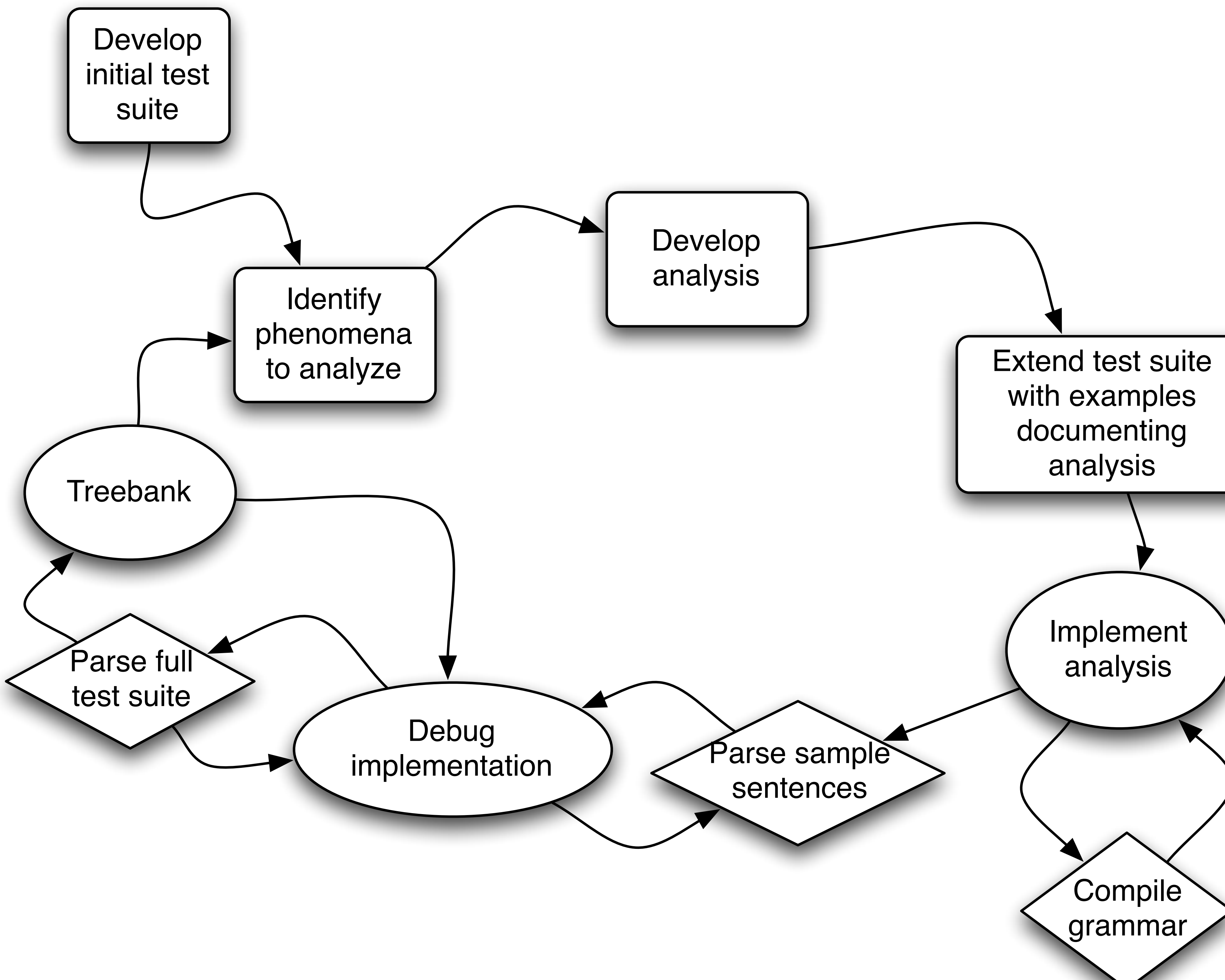
ERG: Examples

INDEX: e2
RELS:
h1:subord_rel(ARG0: e4,ARG1: h5,ARG2: h6)
h7:"_embarassed/JJ_u_unknown_rel"(ARG0: e8,ARG1: i9)
h7:_over_p_rel(ARG0: e10,ARG1: e8,ARG2: x11)
h12:udef_q_rel(ARG0: x11,RSTR: h13,BODY: h14)
h15:nominalization_rel(ARG0: x11,ARG1: h16)
h16:"_let_v_1_rel"(ARG0: e17,ARG1: i18,ARG2: h19)
h20:pron_rel(ARG0: x21)
h22:pronoun_q_rel(ARG0: x21,RSTR: h23,BODY: h24)
h25:"_catch_v_1_rel"(ARG0: e26,ARG1: i27,ARG2: x21,ARG3: h28)
h25:parg_d_rel(ARG0: e29,ARG1: e26,ARG2: x21)
h30:_on_p_rel(ARG0: e31,ARG1: x21,ARG2: x32)
h33:_the_q_rel(ARG0: x32,RSTR: h34,BODY: h35)
h36:"_verge_n_1_rel"(ARG0: x32)
h36:_of_p_rel(ARG0: e37,ARG1: x32,ARG2: x38)
h39:_such+a_q_rel(ARG0: x38,RSTR: h40,BODY: h41)
h42:"_naïve/JJ_u_unknown_rel"(ARG0: e43,ARG1: x38)
h42:"_untruth_n_1_rel"(ARG0: x38)
h44:pron_rel(ARG0: x3)
h45:pronoun_q_rel(ARG0: x3,RSTR: h46,BODY: h47)
h48:"_cough_v_1_rel"(ARG0: e2,ARG1: x3)
h48:loc_nonsp_rel(ARG0: e49,ARG1: e2,ARG2: x50)
h51:udef_q_rel(ARG0: x50,RSTR: h52,BODY: h53)
h54:card_rel(CARG: "2",ARG0: e56,ARG1: x50)
h57:_or_c_rel(ARG0: e58,L-INDEX: e56,R-INDEX: e59,L-HNDL: h54,R-HNDL: h60)
h60:card_rel(CARG: "3",ARG0: e59,ARG1: x50)
h57:"_times_n_1_rel"(ARG0: x50)
h62:"_in+order+to_x_rel"(ARG0: e63,ARG1: h64,ARG2: h65)
h66:"_put_v_1_rel"(ARG0: e67,ARG1: x3,ARG2: x68,ARG3: h69)
h70:_the_q_rel(ARG0: x68,RSTR: h71,BODY: h72)
h73:"_little_a_1_rel"(ARG0: e74,ARG1: x68)
h73:"_prince_n_of_rel"(ARG0: x68,ARG1: i75)
h76:_in_p_rel(ARG0: e77,ARG1: x68,ARG2: x78)

Pen and paper syntax work-flow



Grammar engineering work flow (Bender et al 2011)



Main claims

- Grammar engineering allows us to off-load the tedious part of verifying analyses to a computer
- The Grammar Matrix customization system speeds up the process of grammar engineering
    - … while also providing a testbed for typological generalizations
- Grammar engineering can be useful in work with endangered and other understudied languages
- We can automate the first steps of grammar development by inferring answers to the Grammar Matrix questionnaire from IGT
    - … and this process itself provides useful insight into data collections

Overview

- Grammar engineering
- The LinGO Grammar Matrix
- AGGREGATION

LinGO Grammar Matrix: Motivations and early history

- Speed up grammar development
  - Initial context: Project DeepThought
  - Leverage resources from resource-rich language to enhance NLP for resource-poor languages
  - Claim: Some of what was learned in ERG development is not English-specific
- Interoperability: a family of grammars compatible with the same downstream processing tools

Grammar Matrix: Motivations and early history

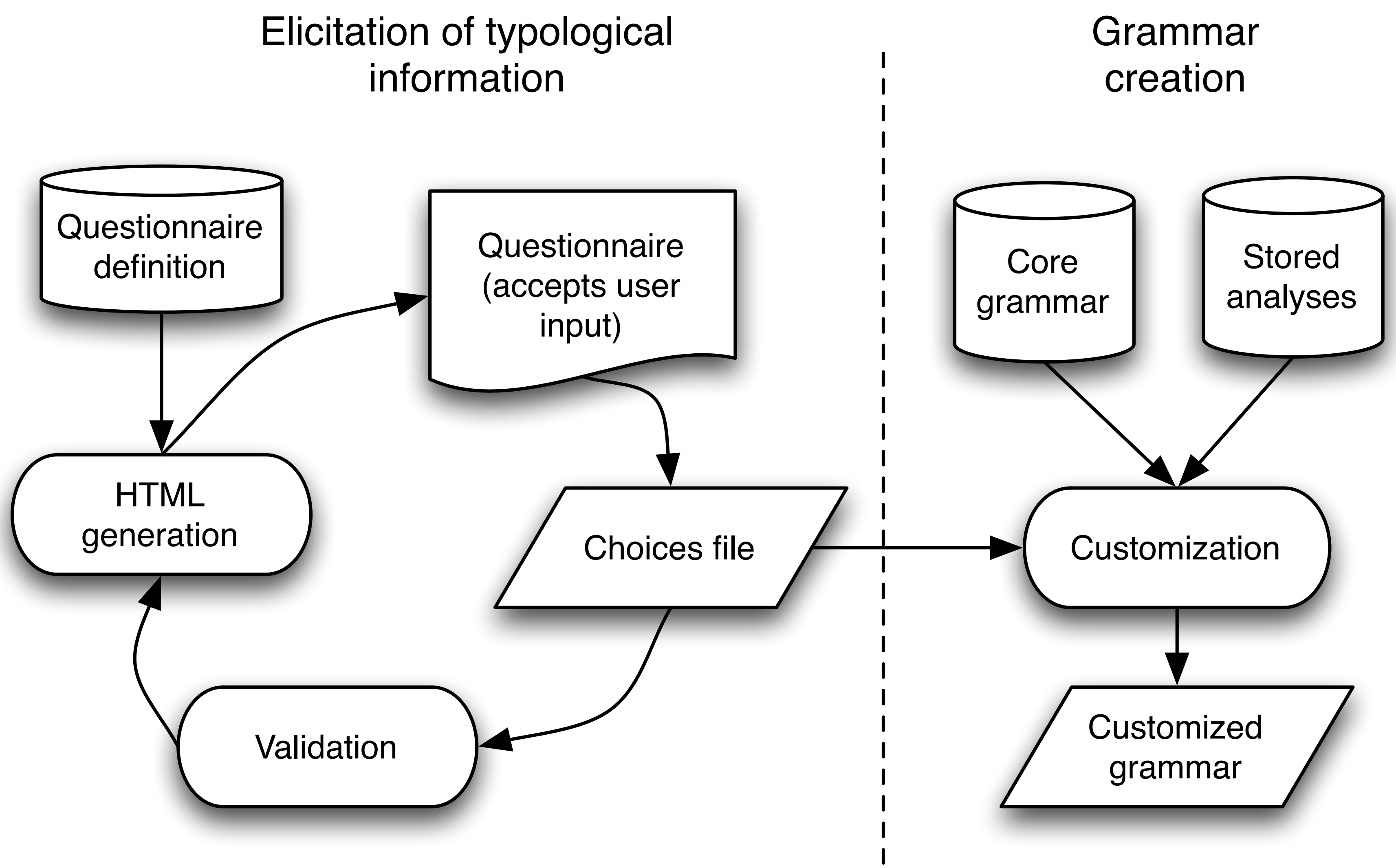
- With reference to Jacy (Siegel et al 2016), strip everything from ERG (Flickinger 2000, 2011) which looks English-specific
- Resulting “core grammar” doesn’t parse or generate anything, but supports quick start-up for scaleable resources (Bender et al 2002)
- Used in the development of grammars for Norwegian (Hellan & Haugereid 2003), Modern Greek (Kordoni & Neu 2005), Spanish (Marimon 2010) and Italian
- Used as the basis of multilingual grammar engineering course at UW (Ling 567): 118 languages since 2004

Grammar customization: Motivations

- The Grammar Matrix core grammar is not itself a functioning grammar fragment
  - can't be directly tested
- Human languages vary along many dimensions, but not infinitely
- Can be seen as solving many of the same problems in different ways
- Many phenomena are "widespread, but not universal" (Drellishak, 2009)
  - we can do more than refining the core
- Also, grammar engineering lab instructions started getting mechanistic

LinGO Grammar Matrix Customization System

(Bender & Flickinger 2005, Drellishak 2009, Bender et al 2010)



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

- ? General Information
- Word Order
- Number
- Person
- Gender
- Case
- Adnominal Possession
- Direct-inverse
- Tense, Aspect and Mood
- Evidentials
- Other Features
- Sentential Negation
- Coordination
- Matrix Yes/No Questions
- Information Structure
- Argument Optionality
- Nominalized Clauses
- Clausal Complements
- Clausal Modifiers
- Lexicon
- Morphology
- Import Toolbox Lexicon
- Test Sentences
- Test by Generation Options

Archive type: (•) .tar.gz ( ) .zip

[Create Grammar] [Test by Generation]

▼ neg-prefix (verb-pc1)

Verb Position Class 1:

Position Class Name: neg-prefix

Obligatorily occurs: ☐

Appears as a prefix or suffix: prefix

Possible inputs: endoclitics (verb-pc44) ▼

Morphotactic Constraints:

Add a Require constraint

Add a Forbid constraint

Lexical Rule Types that appear in this Position Class:

▶ neg (verb-pc1_lrt1)

▼ finite-neg (verb-pc1_lrt2)

Lexical Rule Type 2:

Name: finite-neg

Supertypes: neg (verb-pc1_lrt1) ▼

Features:

× Name: form Value: finite ▼ Specified on: the verb

Add a Feature

Morphotactic Constraints:

× Lexical Rule Type 2 requires one of the following: verb-pc5_lrt1, verb-pc5_lrt ▼

Add a Require constraint

Current and near-future libraries (1/2)

- Word order (Bender & Flickinger 2005, Fokkens 2010)
- Coordination (Drellishak & Bender 2005)
  - Agreement in coordination (Dermer ms)
- Matrix yes-no questions* (Bender & Flickinger 2005)
- Morphotactics (O'Hara 2008, Goodman 2013)
- Case (+ direct-inverse marking) (Drellishak 2009)
- Agreement (person, number, gender) (Drellishak 2009)
- Argument optionality (pro-drop) (Saleem & Bender 2010)
- Tense and aspect (Poulson 2011)
- Sentential negation (Bender & Flickinger 2005, Crowgey 2012)

Current and near-future libraries (2/2)

- Information structure (Song 2014)
- Adjectives (attributive, predicative, incorporated) (Trimble 2014)
- Evidentials (Haeger7)
- Valence alternations (Curtis 2018)
- Adnominal possessives (Nielsen 2018)
- Adverbial clauses (Howell & Zamaraeva 2018)
- Clausal complements (Zamaraeva et al to appear)
- Nominalization (Howell et al 2018)
- *Wh*- questions (Zamaraeva in progress)

Creating a library for the customization system

- Choose phenomenon
- Review typological literature on phenomenon
- Refine definition of phenomenon
- Conceptualize range of variation within phenomenon
- Review HPSG (& broader syntactic) literature on phenomenon
- Pin down target MRSs
- Develop HPSG analyses for each variant
- Implement analyses in tdl
- Develop questionnaire
- Run regression tests
- Test with pseudo-languages
- Test with illustrative languages
- Test with held-out languages
- Add tests to regression tests
- Add to MatrixDoc pages

Main claims

- Grammar engineering allows us to off-load the tedious part of verifying analyses to a computer
- The Grammar Matrix customization system speeds up the process of grammar engineering
  - … while also providing a testbed for typological generalizations
- Grammar engineering can be useful in work with endangered and other understudied languages
- We can automate the first steps of grammar development by inferring answers to the Grammar Matrix questionnaire from IGT
  - … and this process itself provides useful insight into data collections

Overview

- Grammar engineering
- The LinGO Grammar Matrix
- AGGREGATION

Evaluation: Is the Matrix an effective starting point for Grammar Engineering? (Bender 2008)

Evaluation: Is the Matrix an effective starting point for Grammar Engineering? (Bender 2008)

- Wambaya [wmb] is a Mirndi language spoken in the West Barkly Tablelands region (Northern Territory) of Australia (Nordlinger 1998; Green & Nordlinger 2004)

Evaluation: Is the Matrix an effective starting point for Grammar Engineering? (Bender 2008)

- Wambaya [wmb] is a Mirndi language spoken in the West Barkly Tablelands region (Northern ... en & Nordlinger 2004)

Evaluation: Is the Matrix an effective starting point for Grammar Engineering? (Bender 2008)

- Wambaya [wmb] is a Mirndi language spoken in the West Barkly Tablelands region (Northern Territory) of Australia (Nordlinger 1998; Green & Nordlinger 2004)

Evaluation: Is the Matrix an effective starting point for Grammar Engineering? (Bender 2008)

- Wambaya [wmb] is a Mirndi language spoken in the West Barkly Tablelands region (Northern Territory) of Australia (Nordlinger 1998; Green & Nordlinger 2004)

- Developed by hand (with jump start from the Grammar Matrix) on the basis of the analyses in Nordlinger 1998 (incl. IGT)

Evaluation: Is the Matrix an effective starting point for Grammar Engineering? (Bender 2008)

- Wambaya [wmb] is a Mirndi language spoken in the West Barkly Tablelands region (Northern Territory) of Australia (Nordlinger 1998; Green & Nordlinger 2004)
- Developed by hand (with jump start from the Grammar Matrix) on the basis of the analyses in Nordlinger 1998 (incl. IGT)
- Development set: 794 examples from Nordlinger 1998

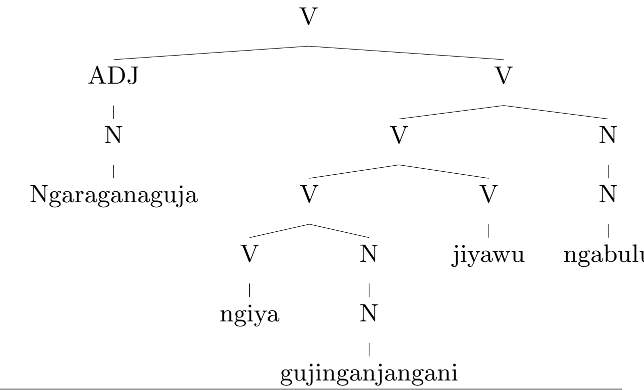
Evaluation: Is the Matrix an effective starting point for Grammar Engineering? (Bender 2008)

- Wambaya [wmb] is a Mirndi language spoken in the West Barkly Tablelands region (Northern Territory) of Australia (Nordlinger 1998; Green & Nordlinger 2004)

- Developed by hand (with jump start from the Grammar Matrix) on the basis of the analyses in Nordlinger 1998 (incl. IGT)

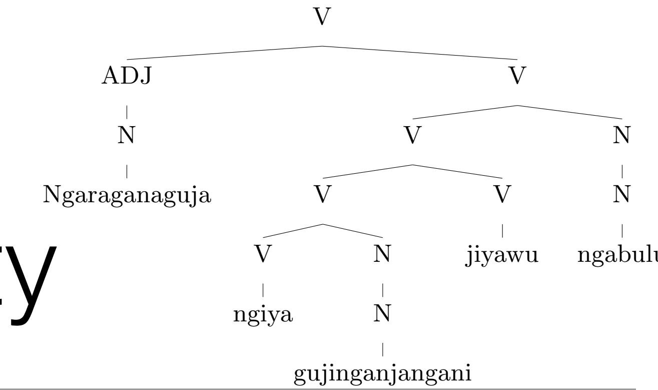
- Development set: 794 examples from Nordlinger 1998

- Held-out test data: Narrative from Nordlinger 1998



Wambaya grammar scope

- Word order (2nd position auxiliary, discontinuous noun phrases)
- Argument optionality
- Linking of syntactic to semantic arguments
- Case (split ergativity)
- Agreement (verb-subject, verb-object, adj-noun)
- Lexical adverbs, including manner, time, location, and negation
- Derived event modifiers
- Derived nominal modifiers
- Lexical adjectives (demonstratives, possessives, numerals, others)
- Subordinate clauses (clausal complements, purposives, simultaneous and prior events)
- Verbless clauses: adjective, nouns, and adverbs functioning as predicates
- Illocutionary force: imperatives, declaratives and interrogatives, including wh-questions
- Coordination: of clauses and noun phrases
- Inalienable possession construction
- Secondary predicates
- Causatives of verbs and adjectives



Sample analysis: Non-configurationality

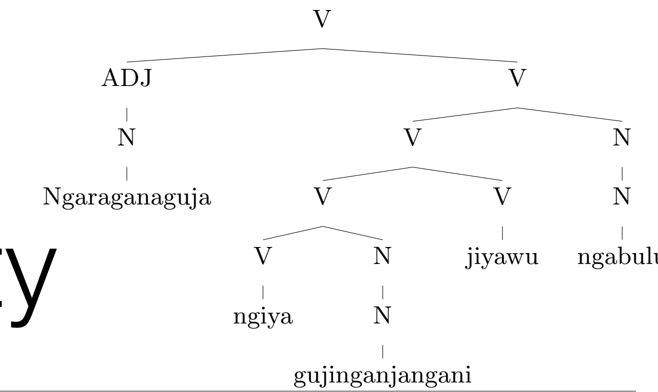
- ‘x7’ is the ARG0 of ‘milk’ and the ARG1 of ‘proprietive’, even though the words contributing these relations are far apart in the sentence.

(1)

Ngaragana-nguja	ngiy-a	gujinganjanga-ni	jiyawu	ngabulu.
grog-PROP.IV.ACC	3.SG.NM.A-PST	mother.II.ERG	give	milk.IV.ACC

‘(His) mother gave (him) milk with grog in it.’ [wmb]

Sample analysis: Non-configurationality



- ‘x7’ is the ARG0 of ‘milk’ and the ARG1 of ‘proprietive’, even though the words contributing these relations are far apart in the sentence.

LTOP h1

INDEX e2 (prop-or-ques, past)

RELS ⟨

[**_grog_n_rel** LBL h3, ARG0 x4 (3, iv)],

[**proprietive_a_rel** LBL h5, ARG0 e6, ARG1 x7 (3, iv), ARG2 x4],

[**_mother_n_rel** LBL h8, ARG0 x9 (3sg, ii)],

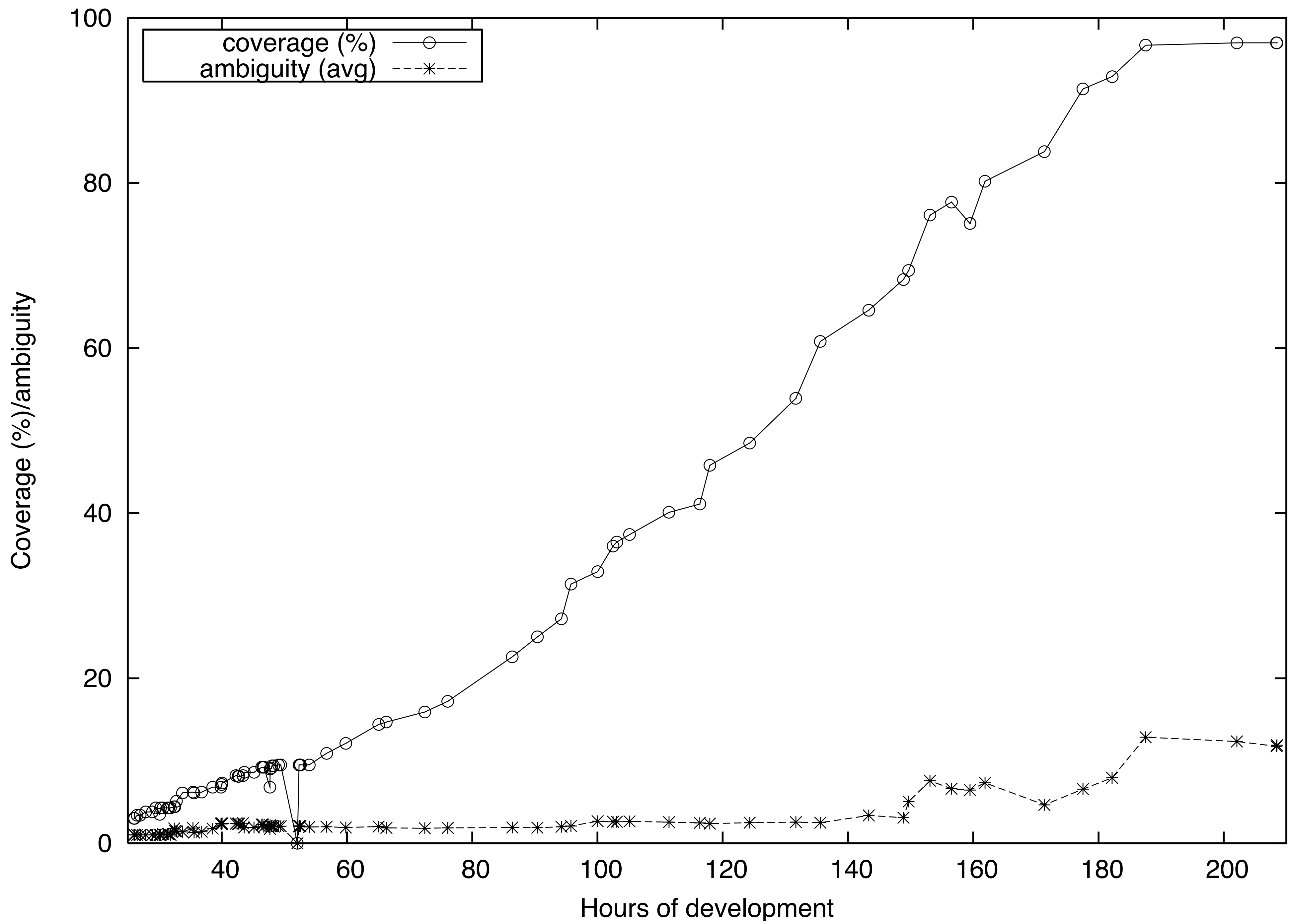
[**_give_v_rel** LBL h1, ARG0 e2, ARG1 x9, ARG2 x10 (3), ARG3 x7],

[**_milk_n_rel** LBL h5, ARG0 x7]

⟩

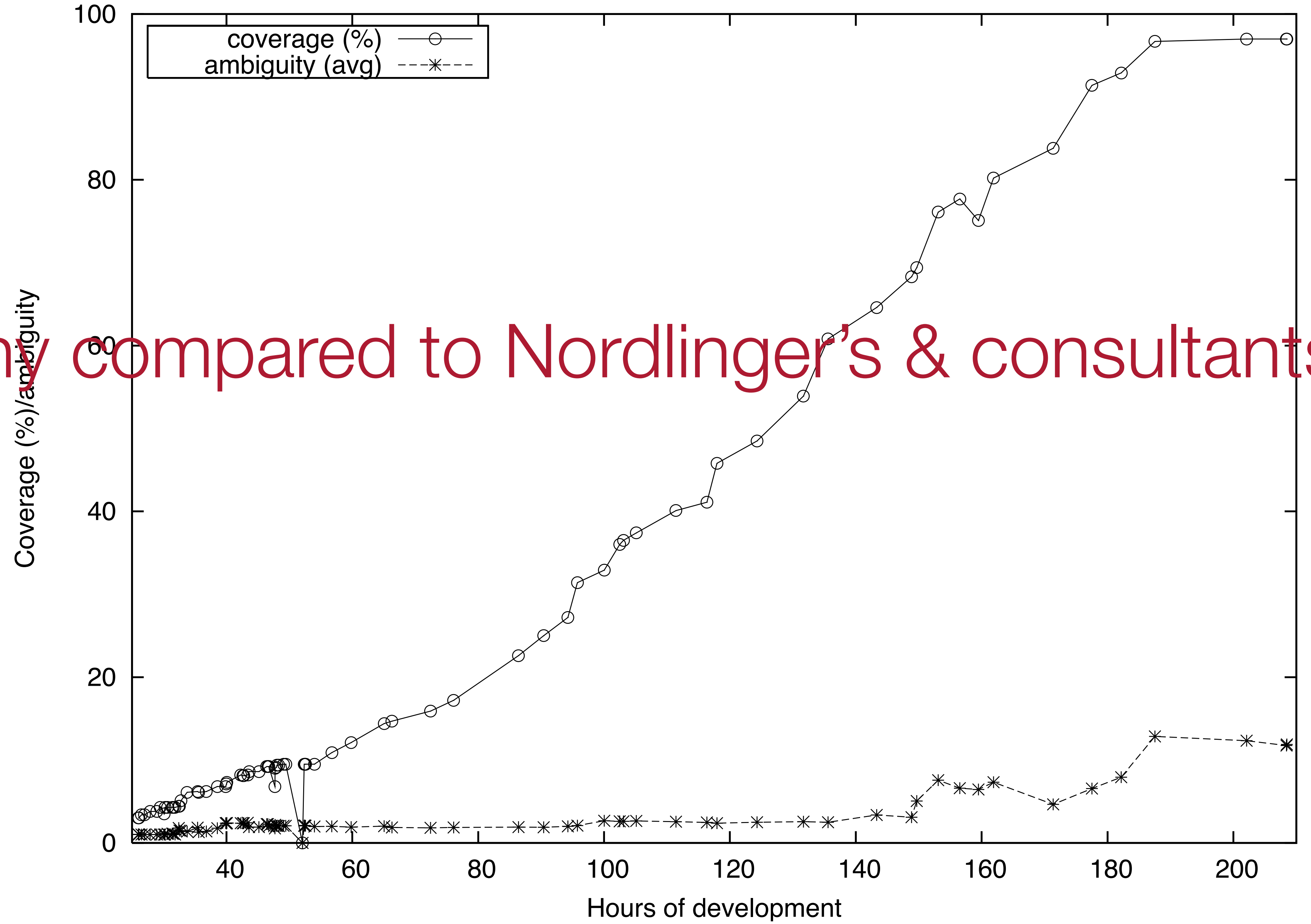
HCONS ⟨ ⟩

Wambaya grammar development



(Bender 2008)

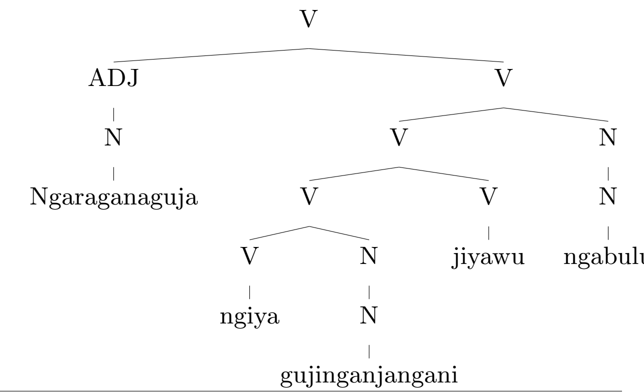
Wambaya grammar development



NB: tiny compared to Nordlinger's & consultants' effort

(Bender 2008)

Wambaya grammar evaluation



- Held out test data “The two Eaglehawks”
- 72 sentences (orig text: 92, removed 20 seen sentences)
- Run twice: before and after adding lexical entries and adjusting morphophonology only

	correct	parsed incorrect	unparsed	average ambiguity
Existing vocab	50%	8%	42%	10.62
w/added vocab	**76%**	8%	14%	12.56

Benefits

- Treebank searchable by syntactic and semantic configurations (Bender et al 2012)
  - Freeform
  - Sample queries embedded in electronic descriptive grammars
- Ability to extend annotations to additional transcribed but unglossed data
- Ability to search both glossed & unglossed data for structures as yet unknown to the grammar (Baldwin et al 2005)
- Long term: Development of language technology

Main claims

- Grammar engineering allows us to off-load the tedious part of verifying analyses to a computer
- The Grammar Matrix customization system speeds up the process of grammar engineering
  - … while also providing a testbed for typological generalizations
- Grammar engineering can be useful in work with endangered and other understudied languages
- We can automate the first steps of grammar development by inferring answers to the Grammar Matrix questionnaire from IGT
  - … and this process itself provides useful insight into data collections

Overview

- Grammar engineering
- The LinGO Grammar Matrix
- AGGREGATION

RiPLes: Leveraging IGT (Xia & Lewis 2007, Lewis & Xia 2008, Xia & Lewis 2009, Georgi 2016)

- Interlinear glossed text (IGT) is an extremely rich data type
- IGT exists in plentiful quantities on the web, even for low resource languages
- Example from Chintang [ctn]:

akka ita khurehẽ

RiPLes: Leveraging IGT (Xia & Lewis 2007, Lewis & Xia 2008, Xia & Lewis 2009, Georgi 2016)

- Interlinear glossed text (IGT) is an extremely rich data type
- IGT exists in plentiful quantities on the web, even for low resource languages
- Example from Chintang [ctn]:

akka ita khurehẽ

'I carried bricks.' [ctn] (Bickel et al., 2012)

RiPLes: Leveraging IGT (Xia & Lewis 2007, Lewis & Xia 2008, Xia & Lewis 2009, Georgi 2016)

- Interlinear glossed text (IGT) is an extremely rich data type
- IGT exists in plentiful quantities on the web, even for low resource languages
- Example from Chintang [ctn]:

akka ita	khurehẽ
akka ita	khur-a-ŋ-e

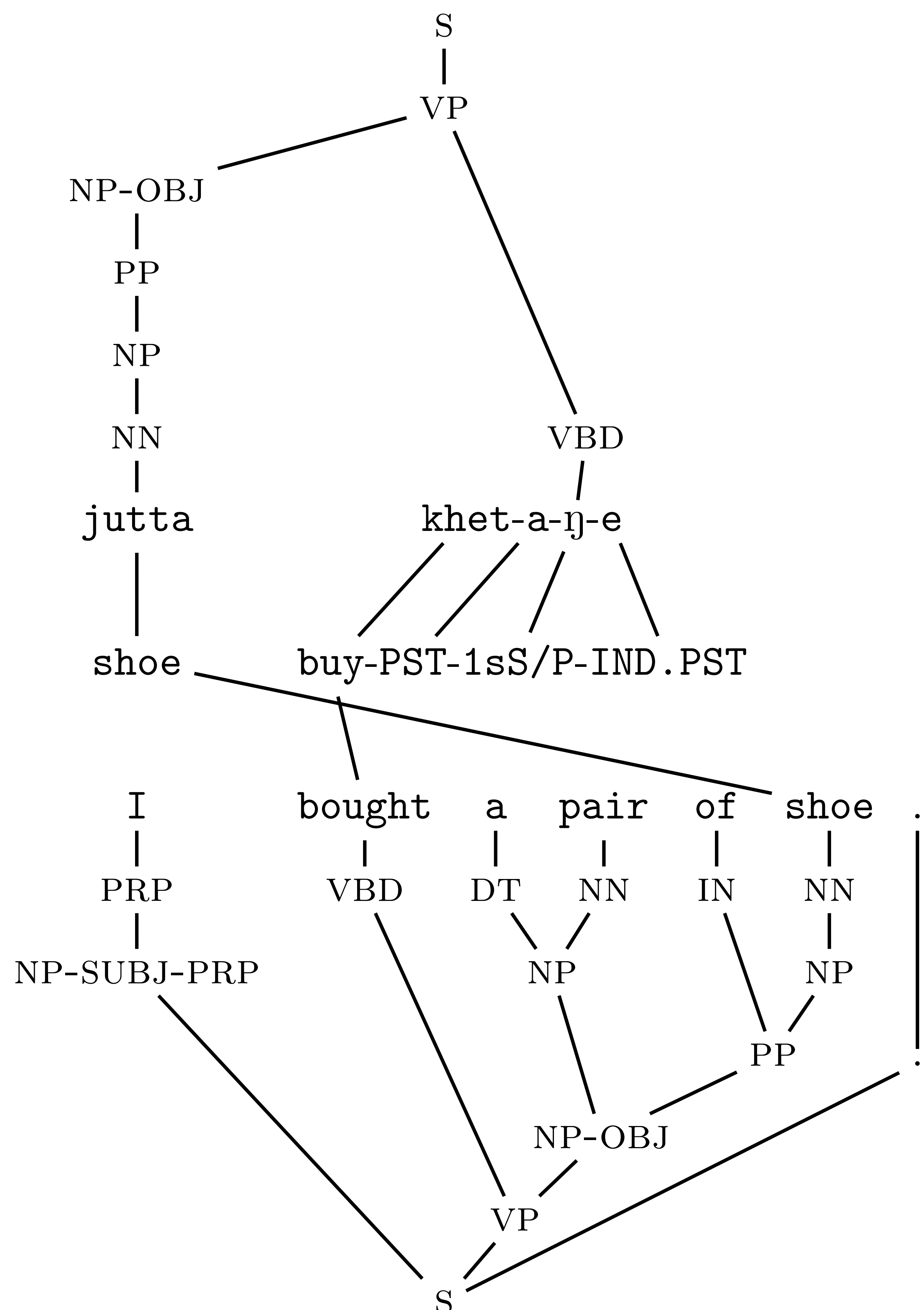
'I carried bricks.' [ctn] (Bickel et al., 2012)

RiPLes: Leveraging IGT (Xia & Lewis 2007, Lewis & Xia 2008, Xia & Lewis 2009, Georgi 2016)

- Interlinear glossed text (IGT) is an extremely rich data type
- IGT exists in plentiful quantities on the web, even for low resource languages
- Example from Chintang [ctn]:

akka	ita	khurehẽ
akka	ita	khur-a-ŋ-e
1s	brick	carry-PST-1sS/P-IND.PST

‘I carried bricks.’ [ctn] (Bickel et al., 2012)



(IGT from Bickel et al 2012)

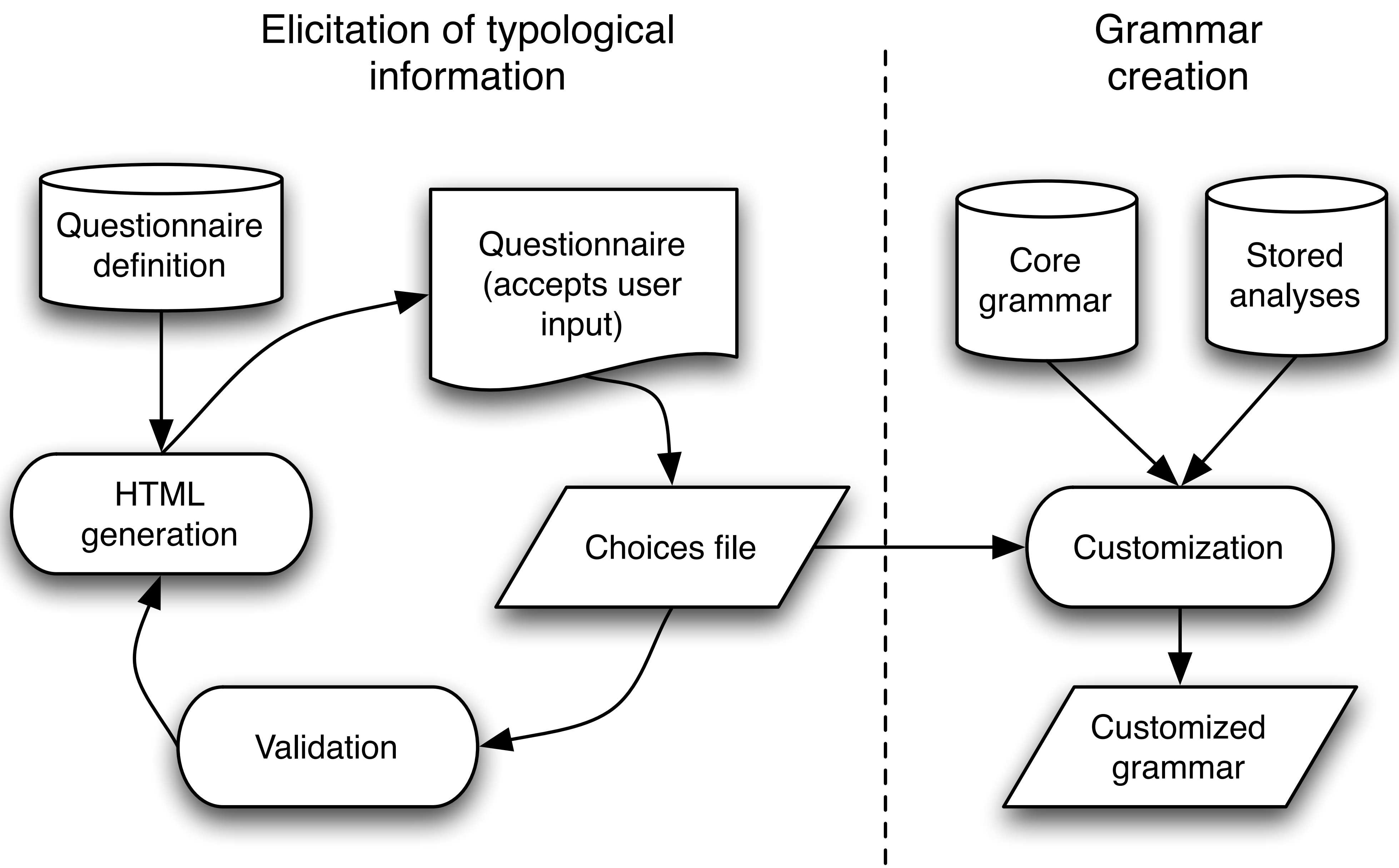
AGGREGATION: Motivation and goals

- Implemented grammars can benefit language documentation
  - Test hypotheses against collected data
  - Sift out unanalyzed phenomena
  - (Eventually) build queryable treebanks (cf. Bender et al 2012)
- But: Implemented grammars are time consuming to build
- Can we leverage the work already done by field linguists to at least start the development of implemented grammars?

Implemented grammars themselves are too complex to be convenient ‘learning targets’

```
neg-lex-rule := neg-prefix-lex-rule-super &
  [ C-CONT [ HOOK [ XARG #xarg,
                    LTOP #ltop,
                    INDEX #ind ],
             RELS <! event-relation &
                     [ PRED "neg_rel",
                       LBL #ltop,
                       ARG1 #harg ] !>,
             HCONS <! qeq &
                      [ HARG #harg,
                        LARG #larg ] !> ],
    SYNSEM.LKEYS #lkeys,
    DTR.SYNSEM [ LKEYS #lkeys,
                 LOCAL [ CONT.HOOK [ XARG #xarg,
                                     INDEX #ind,
                                     LTOP #larg ],
                         CAT.HEAD verb ] ] ].
```

The Grammar Matrix customization system maps simpler descriptions to grammars



▼ neg-prefix (verb-pc1)

Verb Position Class 1:
Position Class Name: neg-prefix
Obligatorily occurs: ☐
Appears as a prefix or suffix: prefix
Possible inputs: endoclitics (verb-pc44) ▼

Morphotactic Constraints:

Add a Require constraint

Add a Forbid constraint

Lexical Rule Types that appear in this Position Class:

▶ neg (verb-pc1_lrt1)
▼ finite-neg (verb-pc1_lrt2)

Lexical Rule Type 2:
Name: finite-neg
Supertypes: neg (verb-pc1_lrt1) ▼

Features:

Name: form Value: finite ▼ Specified on: the verb

Add a Feature

Morphotactic Constraints:

Lexical Rule Type 2 requires one of the following: verb-pc5_lrt1, verb-pc5_lrt ▼

Add a Require constraint

```
section=sentential-negation
infl-neg=on

  verb-pc1_name=neg-prefix
  verb-pc1_order=prefix
  verb-pc1_inputs=verb-pc44
    verb-pc1_lrt1_name=neg
      verb-pc1_lrt1_feat1_name=negation
      verb-pc1_lrt1_feat1_value=plus
      verb-pc1_lrt1_feat1_head=verb
    verb-pc1_lrt2_name=finite-neg
    verb-pc1_lrt2_supertypes=verb-pc1_lrt1
      verb-pc1_lrt2_feat1_name=form
      verb-pc1_lrt2_feat1_value=finite
      verb-pc1_lrt2_feat1_head=verb
      verb-pc1_lrt2_require1_others=verb-pc5_lrt1, verb-pc5_lrt2
      verb-pc1_lrt2_lri1_inflecting=yes
      verb-pc1_lrt2_lri1_orth=mai-
    verb-pc1_lrt3_name=non-finite-neg
    verb-pc1_lrt3_supertypes=verb-pc1_lrt1
      verb-pc1_lrt3_feat1_name=form
      verb-pc1_lrt3_feat1_value=nonfinite
      verb-pc1_lrt3_feat1_head=verb
      verb-pc1_lrt3_lri1_inflecting=yes
      verb-pc1_lrt3_lri1_orth=mai-
```

Bender et al 2013: Inferring large-scale properties
Task 1: Major constituent word order

- Count word order patterns in projected trees
- Calculate ratios of OS:SO etc
- Plot points for each language in 3D space
- Compare to hypothesized canonical points for each word order
- V2 (and not free) if SVO,OVS >> SOV,OSV

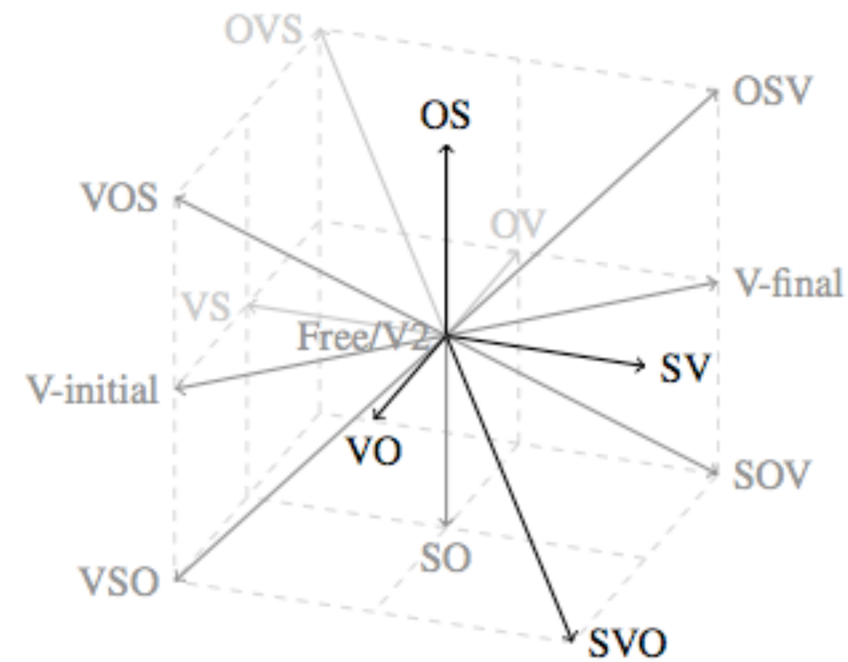


Figure 2: Three axes of basic word order and the positions of canonical word orders.

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

Table 2: Accuracy of word-order inference

Bender et al 2013: Inferring large-scale Task 2: Overall case system

- Method 1 'GRAM': Search for known case grams in gloss line
- Method 2 'SAO': Identify subjects and objects in projected trees; Extract case grams; Compare most frequent case gram for S, A, O

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

Table 3: GRAM case system assignment rules

- Nominative-accusative: S_g=A_g, $S_g \neq O_g$
- Ergative-absolutive: S_g=O_g, $S_g \neq A_g$
- No case: S_g=A_g=O_g, or $S_g \neq A_g \neq O_g$ and S_g, A_g, O_g also present on each of the other argument types
- Tripartite: $S_g \neq A_g \neq O_g$, and S_g, A_g, O_g (virtually) absent from the other argument types
- Split-S: $S_g \neq A_g \neq O_g$, and A_g and O_g are both present in the list for the S argument type

Bender et al 2013: Inferring large-scale Task 2: Overall case system

- Method 1 ‘GRAM’: Search for known case grams in gloss line
- Method 2 ‘SAO’: Identify subjects and objects in projected trees; Extract case grams; Compare most frequent case gram for S, A, O

Bender et al 2013: Inferring large-scale Task 2: Overall case system

- Method 1 ‘GRAM’: Search for known case grams in gloss line
- Method 2 ‘SAO’: Identify subjects and objects in projected trees; Extract case grams; Compare most frequent case gram for S, A, O

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

Table 4: Accuracy of case-marking inference

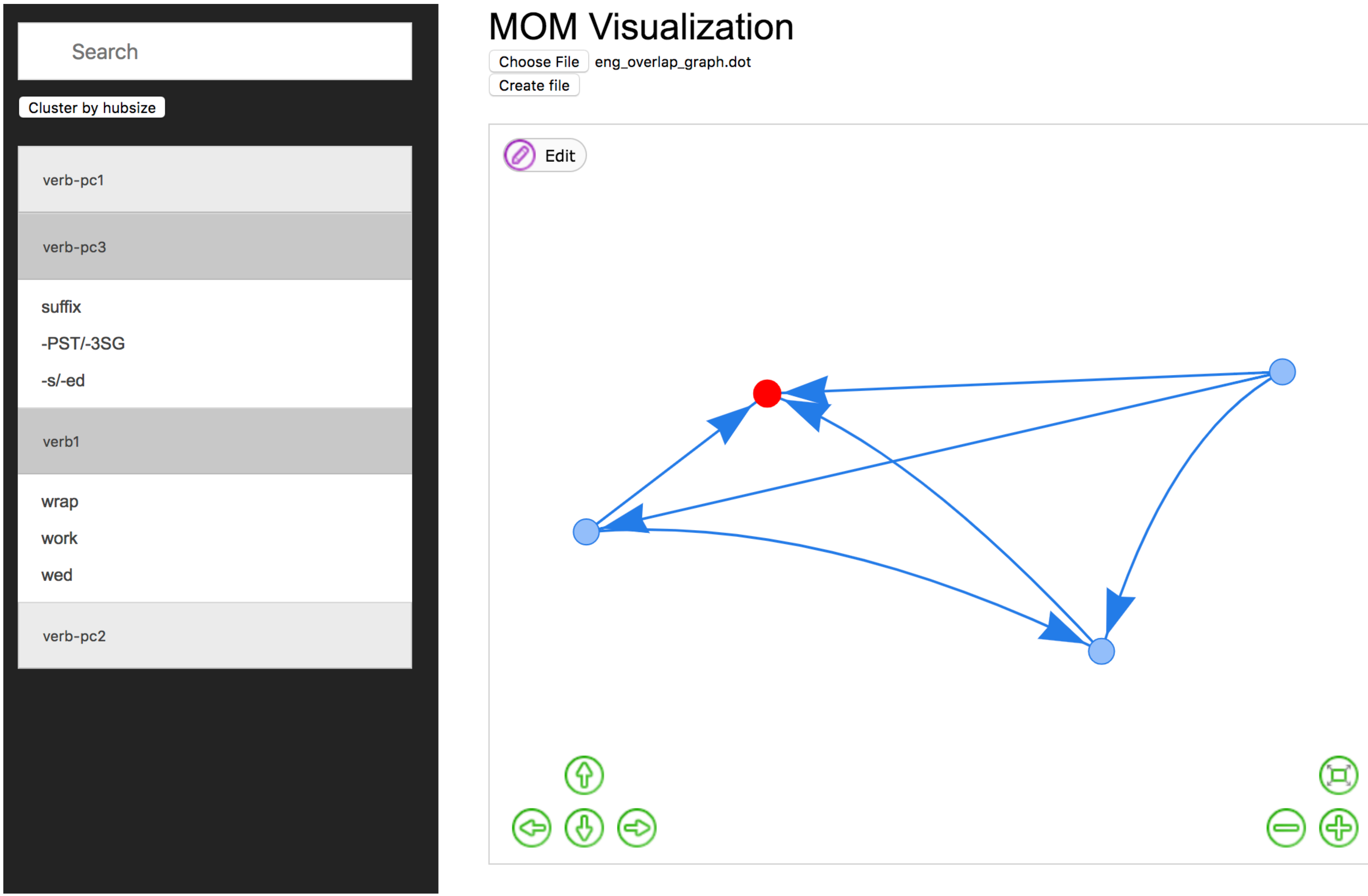
Wax 2014 and Zamaraeva 2016: Learning morphological systems

- Descriptive linguists usually have a pretty good idea of the major constituent word order and case systems of their language
- More time consuming: Morphotactics
  - Create lexical rules for each morpheme, with associated form and morphosyntactic and morphosemantic features
  - Group morphemes into position classes
  - Determine ordering relations

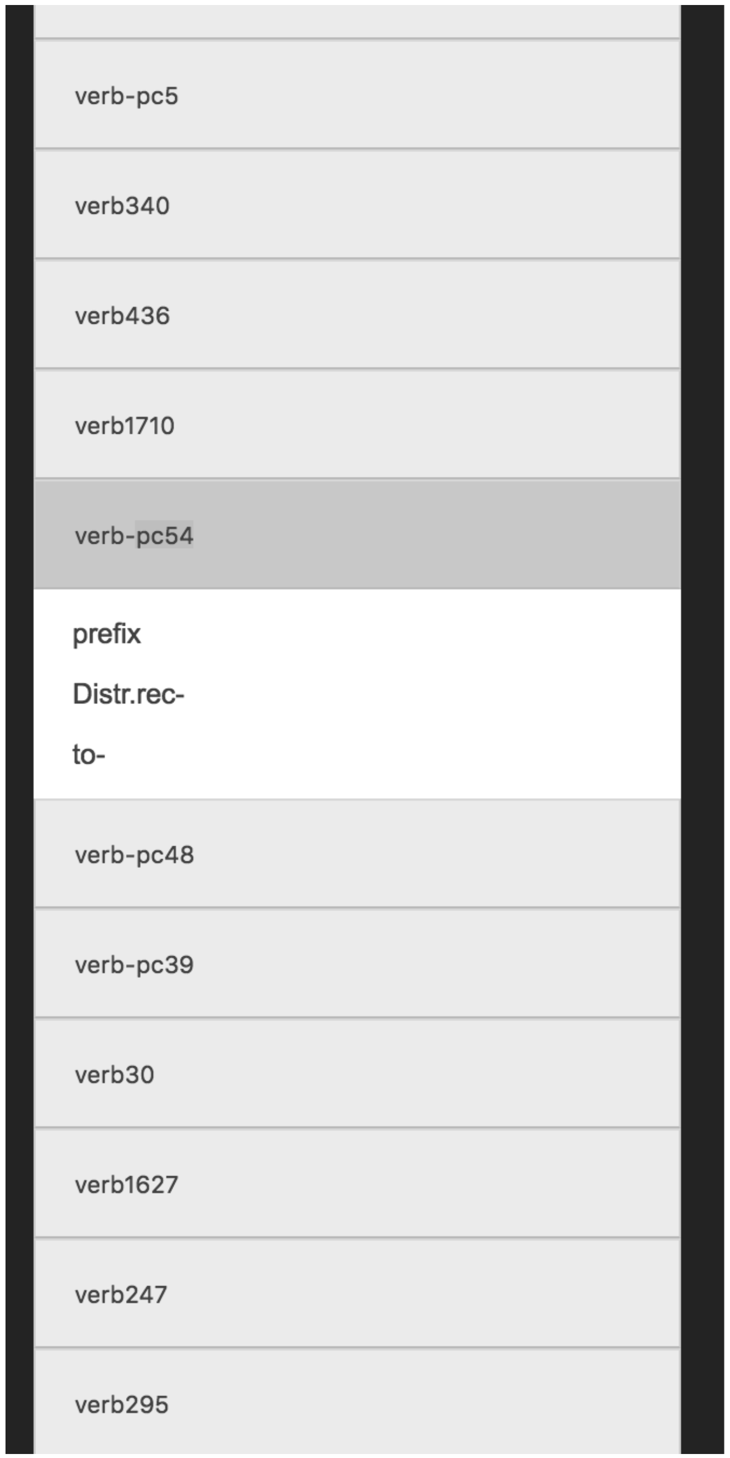
Wax 2014 and Zamaraeva 2016: Learning morphological systems

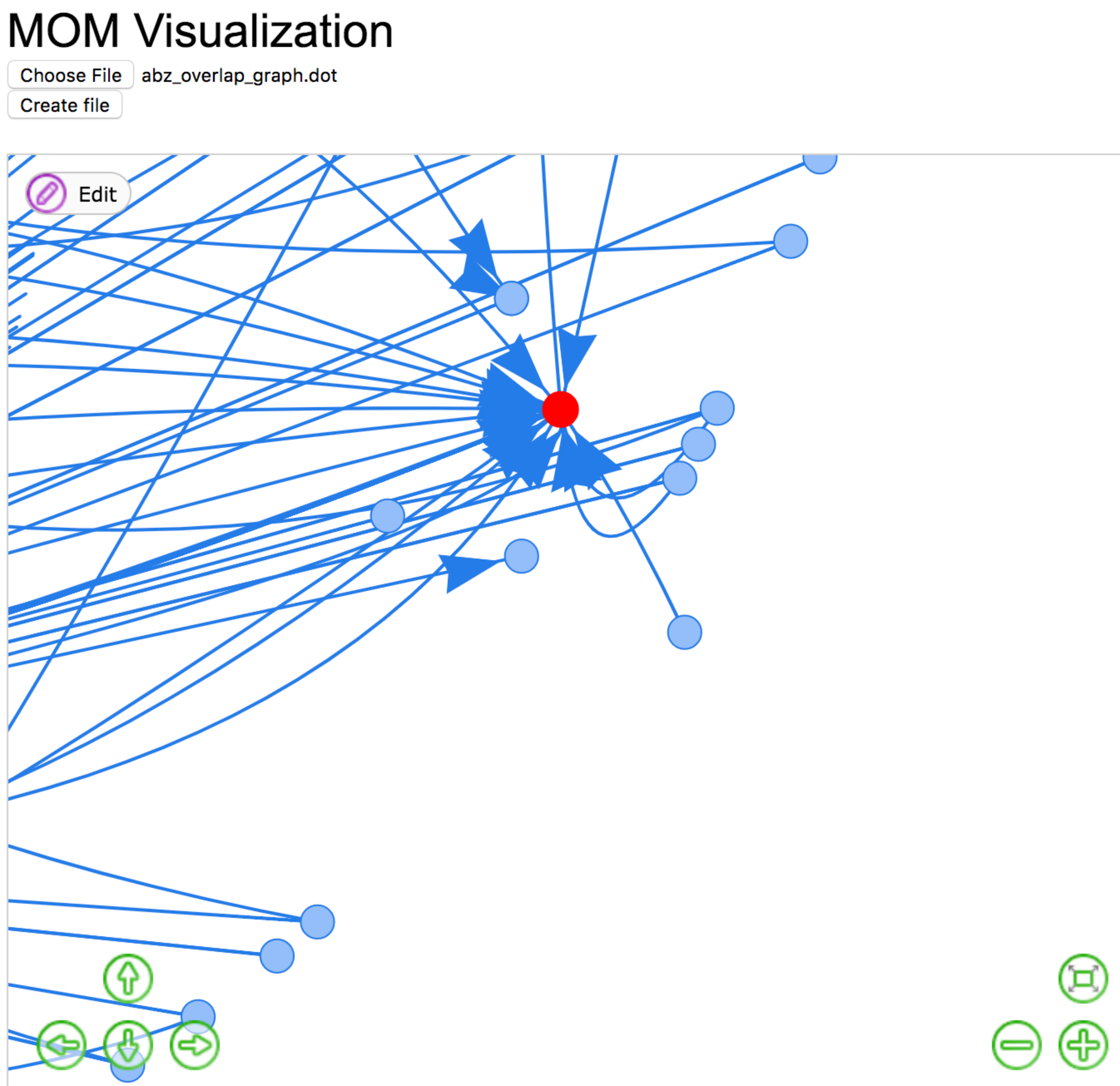
- Start with morpheme-segmented, glossed data
- Observe attested root + affix combinations
- Hypothesize rules attaching affixes to roots or intervening affixes
    - Add features based on grams in glosses
- Combine roots into classes based on affixes that combine with them
    - Tunable parameter: % overlap required
- Combine affixes into position classes based on roots or other affixes they attach to
    - Tunable parameter: % overlap required

Wax 2014 and Zamaraeva 2016: Learning morphological systems



Wax 2014 and Zamaraeva 2016: Learning morphological systems





Bender et al 2014: Working end-to-end prototype

- Answer word order and case system questions
- Extract lexicon (nouns, verbs, determiners)
- Extract morphological rules (for verbs; Wax 2014)
- Extract case values for inflected nouns
- Extract case frames for verbs
- Put in “default” answers for other parts of the questionnaire (e.g. argument optionality)

Case study language: Chintang [ctn]

- An endangered Kiranti language of Nepal (Bickel et al 2007, Schikowski et al 2015)
- (Very) large corpus of high-quality IGT
- Morphologically rich
- Active descriptive research (Bickel et al 2009)

Case study language: Chintang [ctn]

- An endangered Schikowski et al 2015)
- (Very) large corp
- Morphologically
- Active descripti

Learning Case Frames from IGT
(Bender et al 2014)

- For each IGT with overt subject and/or object,
- use grams to find case assigned to subject and object NPs
- no case gram => use default case value for that position

```
# find 1-arg verbs (not dropped subj)
@S < (@VP < @VB=verb !< @OBJ) < @SBJ=sbj !< @OBJ;
# find 1-arg verbs for OSV or VSO langs
@S < @VB=verb < @SBJ=sbj !< @OBJ;

# find 2-arg verbs, both args overt
@S < (@VP < @VB=verb < @OBJ=obj) < @SBJ=sbj;
# 2-arg verbs for OSV or VSO
@S < @VB=verb < @SBJ=sbj < @OBJ=obj;

# find 2-arg verbs w/dropped sbj
@S < (@VP < @VB=verb < @OBJ=obj) !< @SBJ=sbj;
# 2-arg verbs for OSV or VSO
@S < @VB=verb !< @SBJ=sbj < @OBJ=obj;
```

Putting It All Together
(Bender et al 2014)

Choices file	# verb entries	# noun entries	# det entries	# verb affixes	# noun affixes
ORACLE	900	4751	0	160	24
BASELINE	3005	1719	240	0	0
FF-AUTO-NONE	3005	1719	240	0	0
FF-DEFAULT-GRAM	739	1724	240	0	0
FF-AUTO-GRAM	739	1724	240	0	0
MOM-DEFAULT-NONE	1177	1719	240	262	0
MOM-AUTO-NONE	1177	1719	240	262	0

Putting It All Together
(Bender et al 2014)

	Training Data (N = 8863)								Test Data (N = 930)							
choices file	lexical coverage (%)		items parsed (%)		items correct (%)		average readings		lexical coverage (%)		items parsed (%)		items correct (%)		average readings
ORACLE	1165	(13)	174	(3.5)	132	(1.5)	2.17		116	(12.5)	20	(2.2)	10	(1.1)	1.35
BASELINE	1276	(14)	398	(7.9)	216	(2.4)	8.30		41	(4.4)	15	(1.6)	8	(0.9)	28.87
FF-AUTO-NONE	1276	(14)	354	(4.0)	196	(2.2)	7.12		41	(4.4)	13	(1.4)	7	(0.8)	13.92
FF-DEFAULT-GRAM	911	(10)	126	(1.4)	84	(0.9)	4.08		18	(1.9)	4	(0.4)	2	(0.2)	5.00
FF-AUTO-GRAM	911	(10)	120	(1.4)	82	(0.9)	3.84		18	(1.9)	4	(0.4)	2	(0.2)	5.00
MOM-DEFAULT-NONE	1102	(12)	814	(9.2)	52	(0.6)	6.04		39	(4.2)	16	(1.7)	3	(0.3)	10.81
MOM-AUTO-NONE	1102	(12)	753	(8.5)	49	(0.6)	4.20		39	(4.2)	10	(1.1)	3	(0.3)	9.20

Putting It All Together
(Bender et al 2014)

	Training Data (N = 8863)								Test Data (N = 930)							
choices file	lexical coverage (%)		items parsed (%)		items correct (%)		average readings		lexical coverage (%)		items parsed (%)		items correct (%)		average readings	
ORACLE	1165	(13)	174	(3.5)	132	(1.5)	2.17		116	(12.5)	20	(2.2)	10	(1.1)	1.35	
BASELINE	1276	(14)	398	(7.9)	216	(2.4)	8.30		41	(4.4)	15	(1.6)	8	(0.9)	28.87	
FF-AUTO-NONE	1276	(14)	354	(4.0)	196	(2.2)	7.12		41	(4.4)	13	(1.4)	7	(0.8)	13.92	
FF-DEFAULT-GRAM	911	(10)	126	(1.4)	84	(0.9)	4.08		18	(1.9)	4	(0.4)	2	(0.2)	5.00	
FF-AUTO-GRAM	911	(10)	120	(1.4)	82	(0.9)	3.84		18	(1.9)	4	(0.4)	2	(0.2)	5.00	
MOM-DEFAULT-NONE	1102	(12)	814	(9.2)	52	(0.6)	6.04		39	(4.2)	16	(1.7)	3	(0.3)	10.81	
MOM-AUTO-NONE	1102	(12)	753	(8.5)	49	(0.6)	4.20		39	(4.2)	10	(1.1)	3	(0.3)	9.20	

"Though the results are barely measurable in terms of coverage over running text, they nonetheless provide a proof of concept."

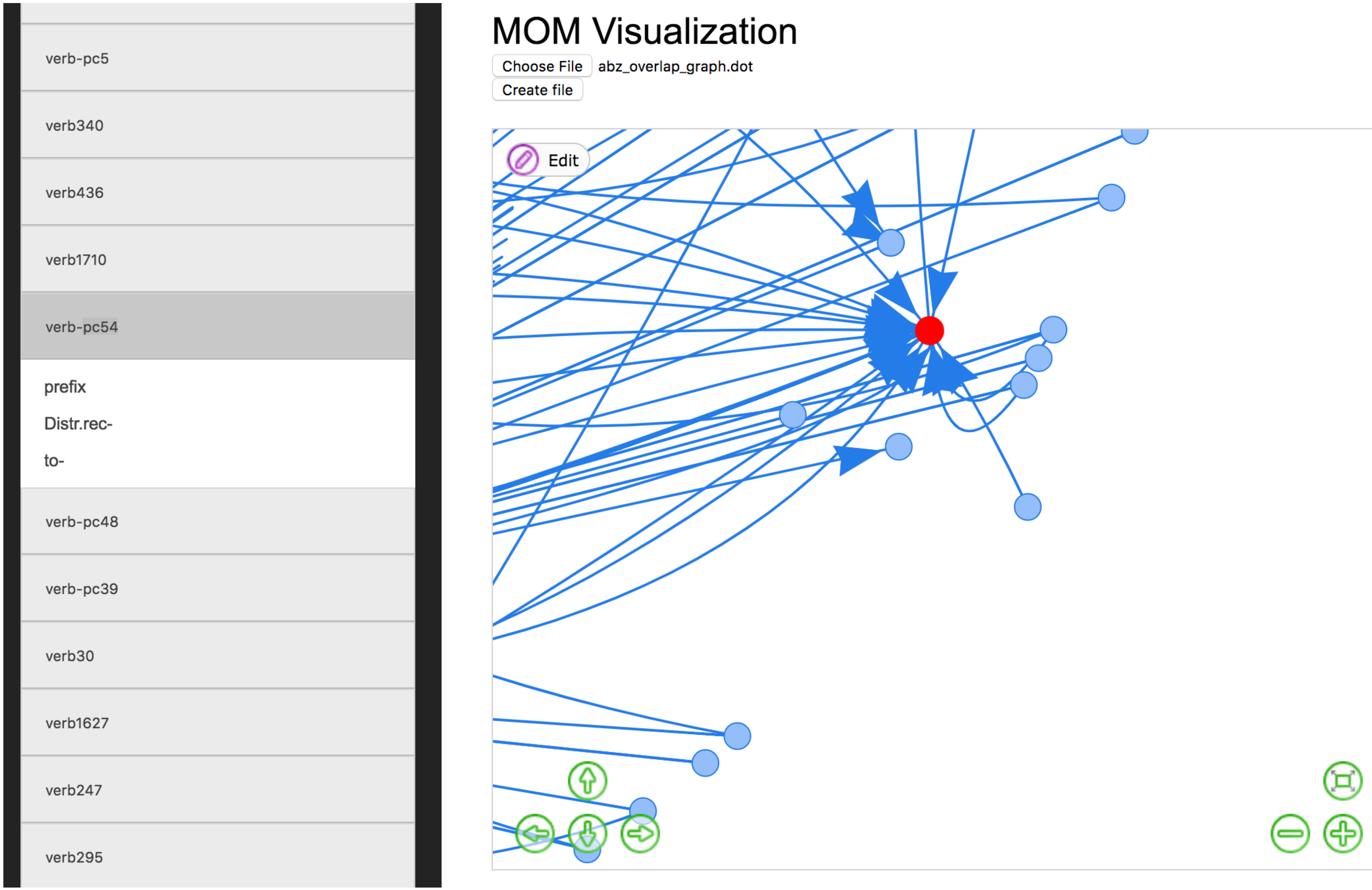
AGGREGATION: Recent developments

- Incorporation of Xigt (Goodman et al 2015) encoding of IGT into AGG pipeline
- Switch over to INTENT (Georgi 2016) for enrichment
  - Provides projected dependency structures
- Extension of morphological extraction to nouns
- Answer more of the questionnaire

What can we offer field linguists now?

- Data overview: Are any of the grams turned into features surprising?
- Consistency checking: Is the IGT well-formatted?
- Consistency checking: Are words glossed consistently?
    - Morphotactics system compresses roots with same spelling & combinatoric potential, even if gloss is different
- Data exploration: Navigating hypothesized position classes, providing feedback to automated system

Morphotactic system exploration



AGGREGATION: Motivation and goals

- Implemented grammars can benefit language documentation
    - Test hypotheses against collected data
    - Sift out unanalyzed phenomena
    - (Eventually) build queryable treebanks (cf. Bender et al 2012)
- But: Implemented grammars are time consuming to build
- Can we leverage the work already done by field linguists to at least start the development of implemented grammars?

Ling 567: Seeking field linguists for collaboration

- Students use the Grammar Matrix to develop grammars for different languages
- Previous model: Three weeks of filling in customization system questionnaire + testsuite development, followed by hand extension of grammars
- Goal for Spring 2019: Start with AGGREGATION inference script's answers to questionnaire & refine, followed by hand extension of grammars
- Requirements: IGT collections and associated sketch grammars for at least 6 languages

Main claims

- Grammar engineering allows us to off-load the tedious part of verifying analyses to a computer
- The Grammar Matrix customization system speeds up the process of grammar engineering
  - … while also providing a testbed for typological generalizations
- Grammar engineering can be useful in work with endangered and other understudied languages
- We can automate the first steps of grammar development by inferring answers to the Grammar Matrix questionnaire from IGT
  - … and this process itself provides useful insight into data collections

References

Adolphs, Peter, Stephan Oepen, Ulrich Callmeier, Berthold Crysmann, Dan Flickinger, and Bernd Kiefer. n.d. Some fine points of hybrid natural language parsing. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*.

Baldridge, Jason, Sudipta Chatterjee, Alexis Palmer, and Ben Wing. 2007. DotCCG and VisCCG: Wiki and programming paradigms for improved grammar engineering with OpenCCG. In T. H. King and E. M. Bender (Eds.), *Proceedings of the GEAF 2007 Workshp*, Stanford, CA. CSLI.

Baldwin, Timothy, John Beavers, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen. 2005. Beauty and the beast: What running a broad-coverage precision grammar over the BNC taught us about the grammar — and the corpus. In S. Kepser and M. Reis (Eds.), *Linguistic Evidence: Empirical, Theoretical, and Computational Perspectives*, 49–69. Berlin: Mouton de Gruyter.

Bateman, John A. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Journal of Natural Language Engineering* 3:15–55.

Bender, Emily M. 2008. Evaluating a crosslinguistic grammar resource: A case study of Wambaya. In *Proceedings of ACL-08: HLT*, 977–985, Columbus, Ohio, June. Association for Computational Linguistics.

Bender, Emily M., Joshua Crowgey, Michael Wayne Goodman, and Fei Xia. 2014. Learning grammar specifications from igt: A case study of chintang. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, 43–53, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation* 1–50. 10.1007/s11168-010-9070-1.

Bender, Emily M., and Dan Flickinger. 2005. Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05 (Posters/Demos)*, Jeju Island, Korea.

Bender, Emily M., 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 J. Carroll, N. Oostdijk, and R. Sutcliffe (Eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 8–14, Taipei, Taiwan.

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2011. Grammar engineering and linguistic hypothesis testing: Computational support for complexity in syntactic analysis. In E. M. Bender and J. E. Arnold (Eds.), *Language from a Cognitive Perspective: Grammar, Usage and Processing*, 5–29. Stanford, CA: CSLI Publications.

Bender, Emily M., Sumukh Ghodke, Timothy Baldwin, and Rebecca Dridan. 2012. From database to treebank: Enhancing hypertext grammars with grammar engineering and treebank search. In S. Nordhoff and K.-L. G. Poggeman (Eds.), *Electronic Grammaticography*, 179–206. Honolulu: University of Hawaii Press.

Bender, Emily M., Michael Wayne Goodman, Joshua Crowgey, and Fei Xia. 2013. Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 74–83, Sofia, Bulgaria, August. Association for Computational Linguistics.

Bergmair, Richard. 2008. Monte Carlo semantics: McPIET at RTE4. In *Text Analysis Conference (TAC 2008) Workshop-RTE-4 Track. National Institute of Standards and Technology.*

Bickel, Balthasar, Goma Banjade, Martin Gaenszle, Elena Lieven, Netra Prasad Paudyal, Ichchha Purna Rai, Manoj Rai, Novel Kishore Rai, and Sabine Stoll. 2007. Free prefix ordering in Chintang. *Language* 83(1):43–73.

Bickel, Balthasar, Martin Gaenszle, Novel Kishore Rai, Elena Lieven, Goma Banjade, Toya Nath Bhatta, Netra Paudyal, Judith Pettigrew, Ichchha P. Rai, Manoj Rai, Robert Schikowski, and Sabine Stoll. 2009. Audiovisual corpus of the Chintang language, including a longitudinal corpus of language acquisition by six children, plus a trilingual dictionary, paradigm sets, grammar sketches, ethnographic descriptions, and photographs.

Bickel, Balthasar, Martin Gaenszle, Novel Kishore Rai, Vishnu Singh Rai, Elena Lieven, Sabine Stoll, G. Banjade, T. N. Bhatta, N Paudyal, J Pettigrew, and M Rai, I. P.and Rai. 2013. Talk of kazi's trip. Accessed: 15 January 2013.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In J. Carroll, N. Oostdijk, and R. Sutcliffe (Eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 1–7.

Callmeier, Ulrich. 2002. Preprocessing and encoding techniques in pet. In S. Oepen, D. Flickinger, J. Tsujii, and H. Uszkoreit (Eds.), *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing.* Stanford, CA: CSLI Publications.

Copestake, Ann. 2002a. Definitions of typed feature structures. In S. Oepen, D. Flickinger, J. Tsujii, and H. Uszkoreit (Eds.), *Collaborative Language Engineering*, 227–230. Stanford, CA: CSLI Publications.

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

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

Crowgey, Joshua. 2012. The syntactic exponence of negation: A model for the LinGO grammar matrix. Master's thesis, University of Washington.

Curtis, Christian Michael. 2018. A parametric implementation of valence-changing morphology in the LinGO Grammar Matrix. Master's thesis, University of Washington.

Doran, Christy, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. 1994. XTAG system: A wide coverage grammar for English. In *Proceedings of the 15th conference on Computational linguistics - Volume 2*, COLING '94, 922–928, Stroudsburg, PA, USA. Association for Computational Linguistics.

Drellishak, Scott. 2009. *Widespread But Not Universal: Improving the Typological Coverage of the Grammar Matrix*. PhD thesis, University of Washington.

Drellishak, Scott, and Emily M. Bender. 2005. A coordination module for a crosslinguistic grammar resource. In S. Müller (Ed.), *The Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar, Department of Informatics, University of Lisbon*, 108–128, Stanford. CSLI Publications.

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.

Flickinger, Dan. 2011. Accuracy v. robustness in grammar engineering. In E. M. Bender and J. E. Arnold (Eds.), *Language from a Cognitive Perspective: Grammar, Usage and Processing*, 31–50. Stanford, CA: CSLI Publications.

Flickinger, Dan, Stephan Oepen, and Gisle Ytrestøl. 2010. WikiWoods. Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta.

Flickinger, Dan, Yi Zhang, and Valia Kordoni. 2012. DeepBank. A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, 85–96, Lisbon, Portugal. Edições Colibri.

Fokkens, Antske S. 2010. Documentation for the Grammar Matrix word order library. Technical report, Saarland University.

Fokkens, Antske Sibelle. 2014. *Enhancing Empirical Research for Linguistically Motivated Precision Grammars*. PhD thesis, Department of Computational Linguistics, Universität des Saarlandes.

Fong, Sandiway. 2014. Unification and efficient computation in the Minimalist Program. In *Language and Recursion*, 129–138. Springer.

Frank, Anette, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg, and Ulrich Schäfer. 2007. Question answering from structured knowledge sources. *Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives* 5:20–48.

Georgi, Ryan. 2016. *From Aari to Zulu: Massively Multilingual Creation of Language Tools using Interlinear Glossed Text*. PhD thesis, University of Washington.

Goodman, Michael Wayne. 2013. Generation of machine-readable morphological rules with human readable input. *UW Working Papers in Linguistics* 30.

Goodman, Michael Wayne, and Emily M. Bender. 2010. What's in a word? redefining the morphotactic infrastructure in the LinGO Grammar Matrix customization system. Unpublished ms., Poster presented at the Morphology and Formal Grammar Workshop at HPSG 2010.

Goodman, Michael Wayne, Joshua Crowgey, Fei Xia, and Emily M Bender. 2015. Xigt: extensible interlinear glossed text for natural language processing. *Language Resources and Evaluation* 49:455–485.

Götz, Thilo, and Walt Detmar Meurers. 1995. Compiling HPSG type constraints into definite clause programs. In *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, Cambridge, MA.

Green, Ian, and Rachel Nordlinger. 2004. Revisiting Proto-Mirndi. In C. Bowern and H. Koch (Eds.), *Australian Languages: Classification and the Comparative Method*, 291–311. Amsterdam: John Benjamins.

Haeger, Michael. 2017. An evidentiality library for the LinGO Grammar Matrix. Master's thesis, University of Washington.

Hellan, Lars, and Petter Haugereid. 2003. NorSource: An exercise in Matrix grammar-building design. In E. M. Bender, D. Flickinger, F. Fouvry, and M. Siegel (Eds.), *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 2003*, 41–48, Vienna, Austria.

Herring, Joshua. 2016. *Grammar Construction in the Minimalist Program.* PhD thesis, Indiana University.

Howell, Kristen, and Olga Zamaraeva. 2018. Clausal modifiers in the Grammar Matrix. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2939–2952.

Howell, Kristen, Olga Zamaraeva, and Emily M. Bender. 2018. Nominalized clauses in the Grammar Matrix. In S. Müller (Ed.), *Proceedings of the 25th International Conference on Head-Driven Phrase Structure Grammar, University of Tokyo.*

Kordoni, Valia, and Julia Neu. 2005. Deep analysis of Modern Greek. In K.-Y. Su, J. Tsujii, and J.-H. Lee (Eds.), *Lecture Notes in Computer Science*, Vol. 3248, 674–683. Berlin: Springer-Verlag.

Kramer, Jared, and Clara Gordon. 2014. Improvement of a Naive Bayes sentiment classifier using MRS-based features. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, 22–29, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Lewis, William D., and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, 685–690, Hyderabad, India.

Marimon, Montserrat. 2010. The Spanish Resource Grammar. In N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias (Eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Müller, Stefan. 2015. The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling* 3(1):21–86.

Nichols, Eric, Francis Bond, Takaaki Tanaka, Sanae Fujita, and Dan Flickinger. 2006. Multilingual ontology acquisition from multiple MRDs. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, 10–17, Sydney, Australia, July. Association for Computational Linguistics.

Nielsen, Elizabeth K. 2018. Modeling adnominal possession in the LinGO Grammar Matrix. Master's thesis, University of Washington.

Nordlinger, Rachel. 1998. *A Grammar of Wambaya, Northern Australia*. Canberra: Research School of Pacific and Asian Studies, The Australian National University.

Oepen, Stephan. 2001. [incr tsdb()] — Competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany.

Oepen, Stephan, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Journal of Research on Language and Computation* 2(4):575 – 596.

Oepen, Stephan, Erik Velldal, Jan Tore Lønning, Paul Meurer, Victoria Rosn, and Dan Flickinger. 2007. Towards hybrid quality-oriented machine translation. On linguistics and probabilities in MT. In *The 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, Skövde, Sweden.

O'Hara, Kelly. 2008. A morphotactic infrastructure for a grammar customization system. Master's thesis, University of Washington.

Packard, Woodley. 2014. UW-MRS: Leveraging a deep grammar for robotic spatial commands. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 812–816, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Packard, Woodley. 2015. Full forest treebanking. Master's thesis, University of Washington.

Pollard, Carl, and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics. Volume 1: Fundamentals*. Chicago, IL and Stanford, CA: Center for the Study of Language and Information. Distributed by The University of Chicago Press.

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

Poulson, Laurie. 2011. Meta-modeling of tense and aspect in a cross-linguistic grammar engineering platform. *UW Working Papers in Linguistics* 28.

Ranta, Aarne. 2007. Modular grammar engineering in GF. *Research on Language & Computation* 5:133–158.

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

Saleem, Safiyyah, and Emily M. Bender. 2010. Argument optionality in the lingo grammar matrix. In *Coling 2010: Posters*, 1068–1076, Beijing, China, August. Coling 2010 Organizing Committee.

Schikowski, Robert, Balthasar Bickel, and Netra Paudyal. 2015. Flexible valency in Chintang. In B. Comrie and A. Malchukov (Eds.), *Valency Classes: A Comparative Handbook*. Berlin: Mouton de Gruyter.

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

Siegel, Melanie, Emily M. Bender, and Francis Bond. 2016. *Jacy: An Implemented Grammar of Japanese*. Stanford CA: CSLI Publications. November.

Slayden, Glenn C. 2012. Array TFS storage for unification grammars. Master's thesis, University of Washington.

Song, Sanghoun. 2014. *A Grammar Library for Information Structure*. PhD thesis, University of Washington.

Stabler, Edward. 2001. Minimalist grammars and recognition. In C. Rohrer, A. Rossdeutscher, and H. Kamp (Eds.), *Linguistic form and its computation*, 327–352. Stanford, CA: CSLI.

Suppes, P., D. Flickinger, B. Macken, J. Cook, and T. Liang. 2012. Description of the EPGY Stanford University online courses for mathematics and language arts. In *International Society for Technology in Education (ISTE) Annual 2012 Conference*, San Diego CA.

Trimble, Thomas James. 2014. Adjectives in the LinGO grammar matrix. Master's thesis, University of Washington.

Wax, David. 2014. Automated grammar engineering for verbal morphology. Master's thesis, University of Washington.

Xia, Fei, and William Lewis. 2007. Multilingual structural projection across interlinearized text. In *NAACL-HLT 2007*, Rochester, NY.

Xia, Fei, and William Lewis. 2009. Applying NLP technologies to the collection and enrichment of language data on the web to aid linguistic research. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education (LaTeCH – SHELT&R 2009)*, 51–59, Athens, Greece, March. Association for Computational Linguistics.

Zamaraeva, Olga. 2016. Inferring morphotactics from interlinear glossed text: Combining clustering and precision grammars. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 141–150.

Zamaraeva, Olga, Kristen Howell, and Emily M. Bender. to appear. Modeling clausal complementation for a grammar engineering resource. To appear in *Proceedings of the 2nd meeting of the Society for Computation in Linguistics*.