The Grammar Matrix and AGGREGATION: Knowledge-Rich NLP for Endangered and Low Resource Languages

Emily M. Bender
University of Washington

First Workshop on Typology for Polyglot NLP
ACL 2019
Florence, Italy
1 August 2019
Acknowledgments


- AGGREGATION collaborators: Fei Xia, Michael Goodman, Joshua Crowgey, David Wax, Olga Zamarava, Ryan Georgi, Kristen Howell, Michael Lockwood, Swetha Ramaswamy, Haley Lepp, Tifa Almeida, Claude Zhang

- 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.
This talk in a nutshell

• Precision grammars model linguistic systems in a machine & human readable form

• The Grammar Matrix facilitates the development of precision grammars
  • by combining the depth of formal syntax with the breadth of typology
  • and provides a mapping from grammar specifications to precision grammars

• We can automatically (largely heuristically) derive grammar specifications from annotations already provided by linguists, with applications to endangered language documentation
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, also Torr et al 2019 (ACL)
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 (Marimón 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), FFBT (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
HPSG in one slide

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
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
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)
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
Minimal Recursion Semantics in one slide

• Key references: Copestake et al 2005, Bender et al 2015
Minimal Recursion Semantics in one slide

• Key references: Copestake et al 2005, Bender et al 2015

• Underspecified description of logical forms
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
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

• Under continuous development since 1993

• Broad-coverage: 85-95% on varied domains: newspaper text, Wikipedia, biomedical 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
• 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 ].
```
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 ],
     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 ] !> ].
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

They forgot to vote.

[Diagram of syntactic tree and dependency graph for the sentence]
ERG: Examples

(Left tree)

S
  PP
    AP
      AP
        Embarassed
        over
      N
        having
        V
        N
          be
          AP
            be
            N
              caught
              N
                of
                DET
                  verge
                  of
                  N
                    such
                    AP
                      naive
                      N
                        untruth.

(Right tree)

S
  NP
    NP
      V
        N
          ADJ
            two
            CONJ
              or
              N
                three
                times,
            to
            V
              N
                ADJ
                  put
                  AP
                    the
                    N
                      prince
                      N
                        wrong.
ERG: Examples

INDEX: e2
RELS:

h1: subord_rel(ARG0: e4, ARG1: h5, ARG2: h6)

h7: "_embarrassed/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_l_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_l_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_l_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: ":naive/JJ_u_unknown_rel"(ARG0: e43, ARG1: x38)

h42: ":untruth_n_l_rel"(ARG0: x38)

h44: pron_rel(ARG0: x3)

h45: pronoun_q_rel(ARG0: x3, RSTR: h46, BODY: h47)

h48: "_cough_v_l_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_l_rel"(ARG0: x50)

h62: "_in+order+to_x_rel"(ARG0: e63, ARG1: h64, ARG2: h65)

h66: "_put_v_l_rel"(ARG0: e67, ARG1: x3, ARG2: x68, ARG3: h69)

h70: the_q_rel(ARG0: x68, RSTR: h71, BODY: h72)

h73: "_little_a_l_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

1. Identify phenomena to analyze
2. Identify key examples
3. Develop analysis
4. Identify cases of interesting predictions
5. Test acceptability of new key examples
6. Refine analysis

The process is iterative, with feedback loops at each step.
Grammar engineering work flow  
(Bender et al 2011)
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): 123 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)
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
**Verb Position Class 1:**
Position Class Name: neg-prefix
Obligatory occurs: 
Appears as a prefix or suffix: prefix
Possible inputs: endocitics (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

**Morphotactic Constraints:**
- Lexical Rule Type 2 requires one of the following: verb-pc5_lrt1, verb-pc5_lrt
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)
- Nominalization (Howell et al 2018)
- Adverbial clauses (Howell & Zamaraeva 2018)
- Clausal complements (Zamaraeva et al 2019)
- 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
- Extend python backend
- 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
<table>
<thead>
<tr>
<th>Library</th>
<th>Citation</th>
<th>Typological sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argument</td>
<td>Saleem (2010); Saleem &amp;</td>
<td>Ackema et al. (eds.) (2006); Dryer (2008)</td>
</tr>
<tr>
<td>Optionality</td>
<td>Bender (2010)</td>
<td></td>
</tr>
<tr>
<td>Tense</td>
<td>Poulson (2011)</td>
<td>Comrie (1985); Dahl (1985); Bybee et al. (1994), <em>inter alia</em></td>
</tr>
<tr>
<td>Aspect</td>
<td>Poulson (2011)</td>
<td>Comrie (1976); Dahl (1985); Bybee et al. (1994), <em>inter alia</em></td>
</tr>
<tr>
<td>Negation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td>Song (2014)</td>
<td>Féry &amp; Krifka (2009); Buring (2010), <em>inter alia</em></td>
</tr>
<tr>
<td>Structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adjectives</td>
<td>Trimble (2014)</td>
<td>Stassen (2003, 2013); Dixon (2004); Dryer (2013a), <em>inter alia</em></td>
</tr>
</tbody>
</table>
Typology and the Grammar Matrix

• Typological surveys provide critical knowledge about the range of variation for specific linguistic phenomena

• Implementation in the Grammar Matrix puts analyses of all of those variants into a system où tout se tien with all of the other implemented phenomena

• Implementation in the Grammar Matrix allows for evaluation on held out languages
AGGREGATION Project: Motivation & overview

• Precision grammars are potentially useful for endangered language documentation (Bender et al 2012)

• Field linguists produce extremely rich annotations in the form of interlinear glossed text

• The Grammar Matrix provides a mapping from grammar specifications to precision grammars

• Can we infer sufficiently accurate and complete grammar specifications from IGT?
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    khurehē

- 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  khurehē

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

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

  ‘I carried bricks.’ [ctn] (Bickel et al., 2012)
I bought a pair of shoes.

(IGT from Bickel et al 2012)
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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Inferred WO</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV1</td>
<td>0.900</td>
<td>0.200</td>
</tr>
<tr>
<td>DEV2</td>
<td>0.500</td>
<td>0.100</td>
</tr>
<tr>
<td>TEST</td>
<td>0.727</td>
<td>0.091</td>
</tr>
</tbody>
</table>

Table 2: Accuracy of word-order inference

• General parameters like word order alone won’t lead to a usable grammar

• Also required: lexicon and morphotactics (and morphophonology…)

  • Create lexical rules for each morpheme, with associated form and morphosyntactic and morphosemantic features

  • Group morphemes into position classes

  • Determine ordering relations

  • Lexicon: part of speech, case frame, argument optionality…
Lepp et al 2019: Visualizing inferred morphotactics
Lepp et al 2019: Visualizing inferred morphotactics
End-to-end evaluation with Chintang [ctn] (Zamareva et al 2019)

<table>
<thead>
<tr>
<th>Choices file</th>
<th># verb entries</th>
<th># noun entries</th>
<th># verb affixes</th>
<th># noun affixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE</td>
<td>899</td>
<td>4750</td>
<td>233</td>
<td>36</td>
</tr>
<tr>
<td>BASELINE</td>
<td>3005</td>
<td>1719</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FF-AUTO-GRAM</td>
<td>739</td>
<td>1724</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MOM-DEFAULT-NONE</td>
<td>1177</td>
<td>1719</td>
<td>262</td>
<td>0</td>
</tr>
<tr>
<td>INTEGRATED</td>
<td>911</td>
<td>1755</td>
<td>220</td>
<td>76</td>
</tr>
</tbody>
</table>

Table 3: Amount of lexical information in each choices file

<table>
<thead>
<tr>
<th>choices file</th>
<th>lexical coverage (%)</th>
<th>parsed (%)</th>
<th>correct (%)</th>
<th>readings</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE</td>
<td>116 (12.5)</td>
<td>20 (2.2)</td>
<td>10 (1.1)</td>
<td>1.35</td>
</tr>
<tr>
<td>BASELINE *</td>
<td>38 (0.4)</td>
<td>15 (1.6)</td>
<td>8 (0.9)</td>
<td>27.67</td>
</tr>
<tr>
<td>FF-AUTO-GRAM</td>
<td>18 (1.9)</td>
<td>4 (0.4)</td>
<td>2 (0.2)</td>
<td>5.00</td>
</tr>
<tr>
<td>MOM-DEFAULT-NONE</td>
<td>39 (4.2)</td>
<td>16 (1.7)</td>
<td>3 (0.3)</td>
<td>10.81</td>
</tr>
<tr>
<td>INTEGRATED</td>
<td>105 (11.3)</td>
<td>32 (3.4)</td>
<td>15 (1.6)</td>
<td>91.56</td>
</tr>
</tbody>
</table>

* We report slightly different results for lexical coverage and average readings for the baseline than Bender et al. (2014) because we removed determiners from the choices file.

Table 4: Results on 930 held-out sentences
Extending inference: Howell (in progress)

- Previously available: major constituent word order, case systems, case frames for verbs, case values for nouns

- Adding: argument optionality, coordination, PNG on nouns and agreeing categories, tense/aspect/mood, sentential negation, adverbial subordinate clauses

- Initial system tested in Ling 567 as starting grammar specifications (noisy!)

- Testing on 15 languages: 5 dev, 5 initial held-out, 5 more held-out
  - Coverage, ambiguity, treebanked accuracy
External resources: WALS &c

• To what extent do the features in WALS map to Grammar Matrix grammar specifications? (Almeida et al 2019)

• Where they do map, what is the best way to leverage them in inference of grammar specifications? (Zhang et al 2019)

• What about AUTOTYP (Bickel & Nichols 2002)?
This talk in a nutshell

• Precision grammars model linguistic systems in a machine & human readable form

• The Grammar Matrix facilitates the development of precision grammars
  • by combing the depth of formal syntax with the breadth of typology
  • and provides a mapping from grammar specifications to precision grammars

• We can automatically (largely heuristically) derive grammar specifications from annotations already provided by linguists, with applications to endangered language documentation
Acknowledgments


