Towards Creating Precision Grammars from Interlinear Glossed Text

Emily M. Bender  Michael W. Goodman
Joshua Crowgey  Fei Xia
{ebender, goodmami, jcrowgey, fxia}@uw.edu

University of Washington

8 August 2013
Motivation:

- Many languages—an important kind of cultural heritage—are dying
- Language documentation takes a lot of time
- Linguists do the hard work and provide IGT, dictionaries, etc.
- Digital resources expand the accessibility and utility of documentation efforts (Nordhoff and Poggeman, 2012)
- Implemented grammars are beneficial for language documentation (Bender et al., 2012)
- We want to automatically create grammars based on existing descriptive resources (namely, IGT)
Example IGT from Shona (Niger-Congo, Zimbabwe)

(1) Ndakanga ndakatenga muchero
    ndi-aka-nga ndi-aka-teng-a mu-chero
    SBJ.1SG-RP-AUX SBJ.1SG-RP-buy-FV CL3-fruit

‘I had bought fruit.’ [sna] (Toews, 2009:34)
Background
The **Grammar Matrix** (Bender et al., 2002; 2010)

- Pairs a core grammar of near-universal types with a repository of implemented analyses
- Customization system transforms high-level description ("choices file") to an implemented HPSG (Pollard and Sag, 1994) grammar
- Customized grammars are ready for further hand-development
- Grammars can be used to parse and generate sentences, giving detailed derivation trees and semantic representations
- Front-end of the customization system is a linguist-friendly web-based questionnaire
Does your language have determiners (as independent words)?  ◯ yes  ◯ no
If so, what is the order of determiners with respect to nouns?  ◯ Noun-Det  ◯ Det-Noun

Does your language have auxiliary verbs?  ◯ yes  ◯ no

If so, please specify the following auxiliary properties:

**Word Order:** Does an auxiliary verb appear before or after its complement?
- ◯ before
- ◯ after

**Complements:** The complements of auxiliaries are:
- ◯ saturated sentences
- ◯ VPs, raising the subject
- ◯ Vs, raising all of its arguments (argument composition)

**Figure:** The Grammar Matrix Questionnaire: Word Order
Figure: The Grammar Matrix Questionnaire: Lexicon
ODIN and RiPLes (Lewis, 2006; Xia and Lewis, 2008)

- RiPLes parses the English line, and projects structure through the gloss line to the original language line.

Figure: Welsh IGT with alignment and projected syntactic structure.
ODIN and RiPLes (continued)

- Xia and Lewis (2008) did typological property inference from CFG rules extracted from projected structures
- **Question**: Can this process be adapted to customize Matrix grammars?
Methodology
Towards automatic grammar creation:

1. Word-order inference (of 10 word order types)
2. Case system inference (of 8 case system types)

Methodology overview:

- Obtain a corpus of IGT for a language
- Find observed (i.e. overt) patterns
- Analyze pattern distributions to infer underlying pattern/system

Data:

- Student-curated testsuites
- Avg 92 sentences per language (min: 11; max: 251)
- Clean and representative, but small

**Question:** The more voluminous/clean/representative the IGT, the better the model?
Word order

- Goal: Infer best word-order choice from projected structure
- Baseline: most frequent word-order (SOV) according to WALS (Haspelmath et al., 2008)
- For each IGT, get a projected parse from RiPLes with functional and part-of-speech tags (SBJ, OBJ, VB)
- Extract observed binary word orders (S/V, O/V, S/O) as relative linear order
- Calculate observed word order coordinates on three axes: SV–VS; OV–VO; SO–OS
- Compare overall observed word-order to canonical word-orders types (SOV, OSV, SVO, OVS, VSO, VOS, V-initial, V-final, V2, Free)
- Select the closest canonical word-order by Euclidean distance
Figure: Three axes of basic word order and the positions of canonical word orders.
## Word-order Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th># lgs</th>
<th>BASELINE</th>
<th>Inferred WO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV1</td>
<td>10</td>
<td>0.200</td>
<td>0.900</td>
</tr>
<tr>
<td>DEV2</td>
<td>10</td>
<td>0.100</td>
<td>0.500</td>
</tr>
<tr>
<td>TEST</td>
<td>11</td>
<td>0.091</td>
<td>0.727</td>
</tr>
</tbody>
</table>

**Table:** Accuracy of word-order inference; BASELINE is ‘SOV’
Error Analysis:

- Noise (e.g. misalignments, non-standard IGT)
- Freer word orders (e.g. most-frequent vs unmarked)
- Unaligned elements (e.g. auxiliaries)
Case Systems—two approaches (and most-freq baseline):

Case-gram presence (GRAM)

- Look for case grams (NOM, ACC, ERG, ABS) on words
- Select system based on presence of certain grams

<table>
<thead>
<tr>
<th>Case system</th>
<th>Case grams present</th>
<th>Top grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>NOM ∨ ERG ∨ ACC</td>
<td>(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 the other argument types</td>
</tr>
<tr>
<td>nom-acc</td>
<td>✓</td>
<td>(S_g=A_g, S_g\neq O_g)</td>
</tr>
<tr>
<td>erg-abs</td>
<td>✓, ✓</td>
<td>(S_g=O_g, S_g\neq A_g)</td>
</tr>
<tr>
<td>split-v</td>
<td>✓, ✓</td>
<td>(S_g\neq A_g\neq O_g, ) and (S_g, A_g, O_g) absent from others</td>
</tr>
<tr>
<td>(conditioned on V)</td>
<td></td>
<td>(S_g\neq A_g\neq O_g, ) and (A_g) and (O_g) both present on S list</td>
</tr>
</tbody>
</table>

Gram distribution (SAO)

- Get gram lists for SBJ or OBJ
  - Transitive: \(A_g, O_g\)
  - Intransitive: \(S_g\)
- Most frequent gram expected to be case-related
Case-system Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th># lgs</th>
<th>BASELINE</th>
<th>GRAM</th>
<th>SAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV1</td>
<td>10</td>
<td>0.400</td>
<td>0.900</td>
<td>0.700</td>
</tr>
<tr>
<td>DEV2</td>
<td>10</td>
<td>0.500</td>
<td>0.900</td>
<td>0.500</td>
</tr>
<tr>
<td>TEST</td>
<td>11</td>
<td>0.455</td>
<td>0.545</td>
<td>0.545</td>
</tr>
</tbody>
</table>

Table: Accuracy of case-marking inference; BASELINE is ‘none’
Error Analysis:

- **GRAM**: Non-standard case grams (e.g. “SBJ”)
- **SAO**: Unaligned elements (e.g. Japanese case markers)
- **SAO**: Top gram not for case (e.g. “3SG”)
- **Both**: Noise (e.g. erroneous annotation)
Conclusion
Summary:

- Language documentation is greatly facilitated with computational resources, including implemented grammars
- We show some first steps at inducing grammars from traditional kinds of resources
  - Inferring word order from projected syntax
  - Inferring case systems from case grams
- Initial results are promising, and informative
- ... but we’re still a long way from producing full grammars
Looking forward:

- Identify and account for noise
- Use larger data sets
- Analyze more phenomena
- Extrinsic evaluation techniques
Thank you!


Grammar Matrix choices file (Maltese):

```
section=word-order
word-order=free
has-dets=yes
noun-det-order=det-noun
has-aux=yes
aux-comp-order=before
aux-comp=v
multiple-aux=no
...

noun8_name=feminine
  noun8_feat1_name=gender
  noun8_feat1_value=fem
noun9_name=m-proper-noun
  noun9_supertypes=noun2, noun3, noun5, noun7
  noun9_feat1_name=person
  noun9_feat1_value=3rd
noun9_det=imp
  noun9_stem1_orth=Pawlu
  noun9_stem1_pred=_named_rel
  noun9_stem2_orth=Ganni
  noun9_stem2_pred=_name_rel
```
Grammar Matrix Libraries

- **Word Order**
  - SOV, OSV, SVO, OVS, VSO, VOS, V-initial, V-final, V2, Free
- **Number**
- **Person**
- **Gender**
- **Case** (and Direct-Inverse)
  - None, Nom-Acc, Erg-Abs, Tripartite
  - Split-S, Fluid-S, Split-V, Split-N, Focus
- **Tense, Aspect, and Mood**
- **Sentential Negation**
- **Coordination**
- **Yes/no questions**
- **Information structure**
- **Argument Optionality**
- **Lexicon and Morphology**
## Data distribution:

<table>
<thead>
<tr>
<th>Sets</th>
<th>DEV1 (n=10)</th>
<th>DEV2 (n=10)</th>
<th>TEST (n=11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size range</td>
<td>16–251</td>
<td>11–229</td>
<td>14–216</td>
</tr>
<tr>
<td>Size median</td>
<td>91</td>
<td>87</td>
<td>76</td>
</tr>
</tbody>
</table>