Implementing Large-Scale LFG Grammar for Wolof

Cheikh Bamba Dione

Department of Linguistic
University of Bergen

November 27, 2012
Project work

1. Build a morphological analyzer for Wolof (spoken in Senegal with \( \approx 10 \) million speakers)

2. Implement a large-scale grammar using the (Lexical Functional Grammar) LFG formalism

- Motivation: No NLP resources available for Wolof

- Parallel Grammar (ParGram) project
  - Aim: produce wide coverage grammars for a variety of languages (English, German, French, Norwegian, Arabic, Urdu, Tigrinya etc.).
  - Collaboratively written grammars within the LFG framework
  - Use of a commonly-agreed-upon set of grammatical features

- NLP development plateforms:
  1. Morphological analysis: Xerox finite state tool (FST)
  2. Parsing: Xerox Linguistic Environment (XLE)
Morphological analysis using the Xerox tool (fst)

1. two-level morphology: 1) a lower surface and 2) an upper or lexical level
2. Input: surface form is transformed into a lexical form (stem + morphosyntactic features)
3. Use of intermediate level
4. The tool handles the input in both directions: analysis and generation

Example

Task: Apply up **fecceekuwaatoon** "untied again" from **fas**: "to tie"

Lexical: fas+V+Base+Inv+E+MPSV+Iter+PST

\[\uparrow\]

Intermediate: fas :i :e :u :aat :oon

\[\uparrow\]

Surface: fecceekuwaatoon
Morphological components

The components of the Wolof FST:

1. Lexicon: contains verbal and nominal stems, ideophone and closed classes
   - Statistics: common nouns (3800), proper nouns (1000), verbs (3500)
2. Morphotactics as **finite-state network** encoding the legal morpheme combination
3. Phonotactics as **finite-state transducers** describing the rules alternation
4. Composition of lexicon + phonotac. into a single network ⇒ **lex. transducer**
- The Wolof Grammar has 95.78 LFG style rules
- Tokenization using FST (handle MWE, clitics, etc.)
- Guessing mechanisms for unknown lexical entries
  1. First guessing strategy: used for words that are recognized by the morphological analyzer but are not in the lexicons.
  2. Second guessing strategy: used for those entries that are not recognized at all.

For modularity, transparency and performance reasons, the lexicons are divided into three lexicons
- A main lexicon containing open classes and which records subcategorization information.
- The second lexicon includes mainly closed class items (stems for determiners, pronouns, prepositions, etc.).
- There is additionally a lexicon for complex predicates entries (morphological applicative, causative, medio-passive etc.).
Robustness Techniques

Special techniques for disambiguation, increasing robustness and coverage

- **FRAGMENT**: the standard grammar collects enough information in cases where an input sentence does not get a full parse.
  - Return-value: well-formed chunks specified as rules in the standard grammar (e.g. NPs, PPs, Ss, etc.) or
  - The individuals input tokens parsed as TOKEN chunks if no chunks are available.

- **SKIMMING**: allows to overcome timeouts and memory problems (has been used to tackle performance problems for the English and German grammar).

- **Disambiguation**:
  - Optimality marks for preferences
  - Using discriminant-based methods
  - Constraint Grammar (CG) Rules
Data description

- Problem for automatic evaluation: no gold-standard available for Wolof.
- Possibility: manual evaluation
- The corpus is collected from stories. The data are randomly split into a development and a test set.

**Table: Development Corpus**

<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of sentences</td>
<td>380</td>
</tr>
<tr>
<td>Total number of words</td>
<td>3875</td>
</tr>
<tr>
<td>Average number of words per sentence</td>
<td>10.0</td>
</tr>
<tr>
<td>Sentences less than 10 words</td>
<td>205</td>
</tr>
<tr>
<td>Sentences between 10 and 15 words</td>
<td>109</td>
</tr>
<tr>
<td>Sentences between 16 and 20 words</td>
<td>44</td>
</tr>
<tr>
<td>Sentences more than 20 words</td>
<td>22</td>
</tr>
<tr>
<td>Description</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Total number of sentences</td>
<td>150</td>
</tr>
<tr>
<td>Total number of words</td>
<td>1439</td>
</tr>
<tr>
<td>Average number of words per sentence</td>
<td>9.0</td>
</tr>
<tr>
<td>Sentences less than 10 words</td>
<td>87</td>
</tr>
<tr>
<td>Sentences between 10 and 15 words</td>
<td>41</td>
</tr>
<tr>
<td>Sentences between 16 and 20 words</td>
<td>16</td>
</tr>
<tr>
<td>Sentences more than 20 words</td>
<td>6</td>
</tr>
</tbody>
</table>
Possible evaluation scheme: classification of errors into minor errors and serious errors.

- Minor errors would include for instance (PP attachment, Scope of coordination, Best solution is not first solution, but among the first 10, pronominal reference, etc.)

- Serious error:
  - Wrong phrase structure in the main clause. This happens when the system builds the wrong tree because it assigns a POS or a subcategorization frame that is wrong in the context.
  - Three or more minor errors