A Bambara Tonalization System for Word Sense Disambiguation Using Differential Coding, Segmentation and Edit Operation Filtering

Luigi (Y.-C.) Liu   Damien Nouvel

ER-TIM, INALCO, 2 rue de Lille, Paris, France

IJCNLP, 2017
1 Introduction

2 Bambara Reference Corpus

3 Related Works

4 System Architecture

5 Methodology
   - Fundamental Definitions
   - Tonalization as Edit Operation
   - Segmentation
   - Edit Operation Filtering

6 Experiment Result

7 Conclusion

8 Perspectives
Introduction

- **Bambara**: african language with 4 tones (´ ` ˇ ˆ)
- **Orthography**: official orthography does not represent tones.
- **Word sense more ambiguous**: some unaccented tokens can correspond to several tonalized forms $\rightarrow$ challenge for NLP applications.
- **Goal**: implement an **automatic tonalizer for Bambara**
  - Improve subsequent NLP processings
  - Facilitate linguistic analysis for Bambara.
Bambara Reference Corpus I

- Bambara Reference Corpus consists of 2 parts:
  - a non-disambiguated subcorpus
  - a man-annotated subcorpus

<table>
<thead>
<tr>
<th>Part</th>
<th>Words (dist.)</th>
<th>Tonalization</th>
<th>Other</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-disamb.</td>
<td>2160M (58M)</td>
<td>38.73%</td>
<td>8.90%</td>
<td>52.35%</td>
</tr>
<tr>
<td>Disamb.</td>
<td>358M (23M)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Corpus statistics

Table: Annotation statistics

Figure: Corpus composition (medium, source)
Wordform annotation is done for each wordform and for 3 main features:

- POS Tagging
- Tone Marker Restoration
- Gloss Assignment

For POS tagging, we use conditional random fields (CRFs; Lafferty et al. 2001) for sequential modeling

- Over 23 morpho-syntactic possible tags
- Accuracy: 94% (satisfying for under-resourced language)

For the tone marker restoration task, we considered using similar methods

- Over 20,870 distinctive tonal forms
- Learning is quiet inefficient due to large-scale label set

Problem: the drawback of modeling sequences of large-scale label set (of tonal form) is the expensive computational cost needed to estimate CRF parameters.
Related Works

- Word-level modeling (Simard (1998), Tufis and Chitu (1990))
  - French Accent insertion: 2-layers Hidden Markov Model (HMM)
  - Romanian Automatic diacritization: 3-gram tagger
- Word and character levels modeling (Elshafei et al. (2006), Scannell (2011), Nguyen et al. (2012))
  - Arabic Diacritization: 1-layer HMM
  - Uni-codification for African languages: Naive Bayes classifier
  - Vietnamese Accent restoration: CRFs and other
- Hybrid approaches (Said et al. (2013), Metwally et al. (2016))
  - CRF + morphological analyzer
  - CRF + HMM + morphological analyzer
- Category Decomposition (Tellier et al., 2010)
  - Decompose label set in smaller pieces to train separately.
  - Result: time-wise efficiency improvement at train phase.
System Architecture

Figure: Block diagram for the proposed Bambara tonalization system at training stage

Figure: Block diagram for the proposed Bambara tonalization system at tonalization stage
Fundamental definitions I

- Discrete random variables
  \[ X \rightarrow \text{non-tonalized token}: \, kelen \]
  \[ Y \rightarrow \text{tonalized token}: \, k\text{èlen} (\text{adj. same}), \, k\text{élén} (\text{intj. already}) \]
  \[ \Delta \rightarrow \text{differential code}: \, (+1, 2, \acute{}), (+1, 2, \acute{})(+1, 4, \acute{}) \]

- Mappings
  \[ \Delta = E(Y; X) \rightarrow \text{encoder function} \]
  \[ Y = D(\Delta; X) \rightarrow \text{decoder function} \]
  \[ Y = D(E(Y; X); X) \]

- Note: predict differential code \( \Delta \), recovery \( Y \) from \( \Delta \) by decoder \( D \).
Tonalization as Edit Operation I

- Code $\Delta$ can be
  - either $\emptyset$ (when $X = Y$, 52.35%)
  - or a concatenation of codewords like $\sigma_1\sigma_2\sigma_3$, ...

- A codeword $\sigma$ is a triplet $(m, p, c)$ containing
  - $m$: operation type (+1 for insertion, -1 for deletion)
  - $p$: position for operation, a positive integer
  - $c$: character (if insertion), $c \in \Omega$

- Encoder $E(y; x) =$ Applying W.-F. algorithm\(^1\) (Wagner and Fischer, 1974) on $(x, y)$ to produce the code $\delta$

- Decoder $D(\delta; x) =$ Applying edit operations in $\delta$ on $x$ to get tonalized token $y$

---

\(^1\)In this article, we apply Wagner-Fischer algorithm in its special case where there are only 2 available edit operations against 3 edit operations including the substitution as in its general case.
To facilitate learning processing, segmentation is introduced to divide data pair \((x, \delta)\) to train in several segments of data pair \((x^{(i)}, \delta^{(i)})\) where \(i\) is segment id.

Learning on segments of data pair is easier because there is less edit operations to predict and this facilitate our tonalization modeling.

The segmentation mode \(w\)

- \(w = -1\) indicates a syllabification (by morphological parser)
- \(w = 0\) for no segmentation
- \(w > 0\) specifies a \(w\)-width regular segmentation\(^2\).

\(^2\)A regular segmenter forms a segment of every \(w\) successive characters, from left to right (i.e. in direction of writing of Bambara), in its input string. By exception, the last segment at output contains the rest of the string which has not yet been segmented so that we allow it to be equal or shorter than a segment of \(w\) characters.
Annotators also introduce: typographic, orthographic corrections.

Focus on tonalization operations → filtering on edit operations.

Tone Marker Filtering: for each position of input string,
  - Remove all insertions except for tone markers
  - Keep only the 1st of tone insertions
  - Keep only the 1st of tone deletions

Edit operation dispatcher $F_m$: it gives from input code $\delta_{in}$ a sub-sequence composed of operations of type $m$, $m = -1, +1$

If $\delta_{in}$ is a filtered result, inverse mapping from $\{F_{-1}(\delta_{in}), F_{+1}(\delta_{in})\}$ to $\delta_{in}$ exists. What we call edit operation assembler.
About half (52.35%) of tokens in BRC do not need any tone markers.

<table>
<thead>
<tr>
<th>Sys.</th>
<th>-1 (Syll.)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority vote</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>0.843</strong></td>
</tr>
<tr>
<td>S ○ E</td>
<td><strong>0.923</strong></td>
<td>0.915</td>
<td><strong>0.922</strong></td>
<td><strong>0.922</strong></td>
<td>0.917</td>
<td>0.893</td>
</tr>
<tr>
<td>time</td>
<td>101.63</td>
<td>25.52</td>
<td>42.03</td>
<td>235.35</td>
<td>378.37</td>
<td>2683.72</td>
</tr>
<tr>
<td>D ○ F ○ S ○ E</td>
<td><strong>0.923</strong></td>
<td>0.912</td>
<td><strong>0.923</strong></td>
<td><strong>0.923</strong></td>
<td>0.918</td>
<td>0.893</td>
</tr>
<tr>
<td>time</td>
<td>19.88</td>
<td>17.62</td>
<td><strong>13.17</strong></td>
<td>15.67</td>
<td>19.62</td>
<td>261.83</td>
</tr>
</tbody>
</table>

**Table:** Accuracy for our system trained with four different system configurations and eight segmentation modes ($p = 50\%$)
Figure: Accuracy and time of training (configurated as $D \circ F \circ S \circ E$ using syllabification) with respect to different training size 90%-10%
### Experiment Result III

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tone Only</td>
<td>58.52%</td>
</tr>
<tr>
<td>Position Only</td>
<td>1.17%</td>
</tr>
<tr>
<td>Tone and Position</td>
<td>0.023%</td>
</tr>
<tr>
<td>Silence</td>
<td>40.08%</td>
</tr>
</tbody>
</table>

**Table:** Error dist. by type for insertion opt. with $p = 50\%$, system = $D \circ F \circ S \circ E$

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>`</td>
<td>0.9541</td>
</tr>
<tr>
<td>`</td>
<td>0.0438</td>
</tr>
<tr>
<td>`</td>
<td>0.0021</td>
</tr>
<tr>
<td>`</td>
<td>0.0000</td>
</tr>
<tr>
<td>`</td>
<td>0.0841</td>
</tr>
<tr>
<td>`</td>
<td>0.9141</td>
</tr>
<tr>
<td>`</td>
<td>0.0015</td>
</tr>
<tr>
<td>`</td>
<td>0.0003</td>
</tr>
<tr>
<td>`</td>
<td>0.0035</td>
</tr>
<tr>
<td>`</td>
<td>0.0322</td>
</tr>
<tr>
<td>`</td>
<td>0.9643</td>
</tr>
<tr>
<td>`</td>
<td>0.0000</td>
</tr>
<tr>
<td>`</td>
<td>0.0952</td>
</tr>
<tr>
<td>`</td>
<td>0.0000</td>
</tr>
<tr>
<td>`</td>
<td>0.9048</td>
</tr>
</tbody>
</table>

**Table:** Confusion matrix on prediction of tone markers
Conclusion

- Differential encoder:
  - Reduce entropy of labels to be predicted, make CRF learning efficient
  - Allow to implement tone marker filter, edit operation decomposition

- Segmentation:
  - Increase tonalization accuracy
  - Greatly reduce training time

- Tone marker filter:
  - Normalize the tonalized token
  - Lead to reduce training time

- Edit operation decomposition unit (dispatcher):
  - Split the tokens in insertion and deletion of tone markers
  - Allows to accelerate furthermore the training time reduction
Take into account more linguistic information for bambara

Generalization for other languages like French, Arabic, Yoruba, etc.

Available resources and tools:

- Bambara Reference Corpus (French):
  http://cormand.huma-num.fr/index.html
- Tonalizer - CRF-based Tone Reconstitution Tool (English):
  https://github.com/vieenrose/tonalizer