Automating question generation and marking of language learning exercises for isiZulu

Nikhil Gilbert and C. Maria Keet

Department of Computer Science, University of Cape Town, South Africa, mkeet@cs.uct.ac.za

CNL’18, Maynooth, Ireland, August 27-28, 2018
Outline

1. Motivation
2. Design
3. Evaluation and discussion
4. Conclusions
Outline

1. Motivation
2. Design
3. Evaluation and discussion
4. Conclusions
Introduction – Language learning

- Exercise books with limited set of questions; practice effect
- Issues with manual marking of homework exercises and tests [Prabitha(2010)]:
  - prone to errors in marking
  - loss of scripts
  - time taken to return the work to students
  - limited options to assess the students’ progression in language learning
- For context in South Africa, isiZulu, in addition:
  - thousands of entry-level isiZulu learners
  - few teachers
- Computer-assisted language learning
- May be useful for, a.o.: more exercises, automated marking
Introduction – Language learning

- Exercise books with limited set of questions; practice effect
- Issues with manual marking of homework exercises and tests [Prabitha(2010)]:
  - prone to errors in marking
  - loss of scripts
  - time taken to return the work to students
  - limited options to assess the students’ progression in language learning
- For context in South Africa, isiZulu, in addition:
  - thousands of entry-level isiZulu learners
  - few teachers
- Computer-assisted language learning
- May be useful for, a.o.: more exercises, automated marking

⇒ How to automate the entry-level exercises and marking isiZulu?
A few features of isiZulu

- Most populous language in SA, first (home) language of $\pm 23\% \ (\geq 10 \text{ million})$
- Member of the Bantu language group, spoken by some 300 million people
- Bantu languages have characteristically agglutinating morphology
- System of noun classes, controls the concordance of all words in a sentence

Abafana abancane bazozithenga izincwadi ezinkulu

*aba*-fana *aba*-ncane *ba*-zo-*zi*-thenga *izi*-ncwadi e-*zi*-nkulu

‘The little boys will buy the big books’
## Noun classes (simplified)

<table>
<thead>
<tr>
<th>NC</th>
<th>Prefix</th>
<th>Examples</th>
<th>NC</th>
<th>Prefix</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>um(u)</td>
<td>umuntu ‘human’</td>
<td>9a</td>
<td>i</td>
<td>ivazi ‘vase’</td>
</tr>
<tr>
<td>2</td>
<td>aba</td>
<td>abantu</td>
<td>9a (6)</td>
<td>ama</td>
<td>amavazi</td>
</tr>
<tr>
<td>1a</td>
<td>u</td>
<td>ugogo ‘grandmother’</td>
<td>9</td>
<td>i(N)</td>
<td>indlovu ‘elephant’</td>
</tr>
<tr>
<td>2a</td>
<td>o</td>
<td>ogogo</td>
<td>10</td>
<td>izi(N)</td>
<td>izindlovu</td>
</tr>
<tr>
<td>3a</td>
<td>u</td>
<td>ushizi ‘cheese’</td>
<td>11</td>
<td>u(lu)</td>
<td>uphawu ‘mark’</td>
</tr>
<tr>
<td>(2a)</td>
<td>o</td>
<td>oshizi</td>
<td>11 (10)</td>
<td>izi(N)</td>
<td>izimphawu</td>
</tr>
<tr>
<td>3</td>
<td>um(u)</td>
<td>umfoloko ‘fork’</td>
<td>14</td>
<td>ubu</td>
<td>ubuhle ‘beauty’</td>
</tr>
<tr>
<td>4</td>
<td>imi</td>
<td>imifoloko</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>i(li)</td>
<td>igama ‘name’</td>
<td>15</td>
<td>uku</td>
<td>ukuhamba ‘to go’</td>
</tr>
<tr>
<td>6</td>
<td>ama</td>
<td>amagama</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>isi</td>
<td>isilwane ‘animal’</td>
<td>17</td>
<td>ku</td>
<td>(locatives)</td>
</tr>
<tr>
<td>8</td>
<td>izi</td>
<td>isilwane</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Related works

- Use grammar banks, like transformation-based grammar exercises (e.g., [Gardent and Perez-Beltrachini(2011)])
  - Very limited documented isiZulu grammar
Related works

- Use grammar banks, like transformation-based grammar exercises (e.g., [Gardent and Perez-Beltrachini(2011)]):
  - Very limited documented isiZulu grammar
- Corpus-based, POS tagged (e.g., [Sinclair(2004)]);
  - Outdated and out-of domain text [Spiegler et al.(2010)]; very limited other POS tagged text
Related works

- Use grammar banks, like transformation-based grammar exercises (e.g., [Gardent and Perez-Beltrachini(2011)])
  - Very limited documented isiZulu grammar
- Corpus-based, POS tagged (e.g., [Sinclair(2004)]);
  - Outdated and out-of domain text [Spiegler et al.(2010)]; very limited other POS tagged text
- NLG for ontology verbalisation; e.g., [Keet and Khumalo(2017)]
  - Few verbalisation algorithms for basics of grammar (plurals, conjugation, negation)
Architecture of the back-end system

Diagram showing the architecture of the back-end system with components such as Language learning exercises, Noun singular, Verb SC modification, Noun plural, Sentence pluralisation, Sentence scrabble, Noun grammar, Sentence generator, Verb grammar, Grammar and NLG library, Noun corpus, Noun chain corpus, Verb chain corpus, Verb roots, General sentences.
Sentence Generator

- Too time consuming to handcraft (very) many sentences
- Basic sentences only, of the patterns `<noun> <verb>` or `<noun> <verb> <noun>` only
- Idea: exploit some of the semantics of the noun class system
Sentence Generator

- Too time consuming to handcraft (very) many sentences
- Basic sentences only, of the patterns `<noun> <verb>` or `<noun> <verb> <noun>` only
- Idea: exploit some of the semantics of the noun class system
- Noun list (n=231), verb list (n=59); terms typical for language learning; e.g., *umfundi* ‘learner’, *ikhaya* ‘home’, *enza* ‘do’, *hamba* ‘go’, *thenga* ‘like’
- Two ‘chain’ lists
  
  nl  | ubaba <1a> washa;sula;faka;khuluma
  nl  | umzali <1;s> ALL.v;e_dumisa;e_cisha
  vl  | washa <t> imoto;umshini;umnyango
  vl  | sula <> ifasitela;imoto;ipuleti
  vl  | khuluma <t> ALL_1;ALL_1a
Reuse of Ontology verbalisation algorithms

- Pluraliser [Byamugisha et al.(2018)] and a new singulariser
  - $u$- / $aba$- (NC1/2) .... $i(n)$- / $izi(n)$- (NC9/10) .... $uku$- (NC15)
- Conjugator (subject concord) [Keet and Khumalo(2014)]
  - $u$- (NC1) .... $in$- (NC9) .... $ku$- (NC15)
- Positive and negative verbs [Keet and Khumalo(2014)];
  - $u$- / $aka$- (NC1) .... $i$- / $ayi$- (NC9) .... $uku$- / $aku$- (NC15), and change final vowel to -$i$ for negative

documentation of the algorithms: http://www.meteck.org/files/geni/
Algorithm 4 (Negation) Verbalisation of negation in an axiom, as disjointness or negated object property (i.e., axioms of type $C \sqsubseteq \neg D$ and $C \sqsubseteq \neg \exists R.D$).

**Require:** $\mathcal{C}$ set of classes, language $\mathcal{L}$ with $\sqsubseteq$ for subsumption and $\neg$ for negation; variables: $A$ axiom, $NC_i$ noun class, $c_1, c_2 \in \mathcal{C}$, $a_1$ term, $a_2$ letter and $n, p$ are concords, $v$ verb stem; functions: $\text{checkNegation}(A)$, $\text{getNSC}(NC_i)$, $\text{getPNC}(NC_i)$.

**Require:** $\text{checkNegation}(A) == \text{true}$

1. if negation directly preceded by $\sqsubseteq$ and directly followed by $c_2$ then
   2. $NC'_1 \leftarrow \text{lookup plural nounclass of } NC_1$
   3. $c'_1 \leftarrow \text{pluralise}(c_1, NC'_1)$
   4. $a_1 \leftarrow \text{lookup quantitative concord for } NC'_1$
   5. $n \leftarrow \text{getNSC}(NC'_1)$
   6. $p \leftarrow \text{getPNC}(NC_2)$
   7. $\text{RESULT} \leftarrow \text{‘}a_1 \; c'_1 \; np \; c_2.\text{’}$

8. else if negation in front of OP then
   9. $v' \leftarrow \text{remove final vowel of } v$
   10. $n \leftarrow \text{getNSC}(NC'_1)$
   11. if $v' \in \{a, e, i, o, u\}$ then
       12. $\text{negv} \leftarrow \text{phonoCondNegSc}(v', n)$
   13. else
       14. $\text{negv} \leftarrow n + v'$
   15. end if
   16. $\text{RESULT} \leftarrow \text{‘}a_1 \; c'_1 \; \text{negv} \; c_2 \; r_2 g_2 \text{dwa.}\text{’}$
   17. else
       $\text{RESULT} \leftarrow \text{‘}\text{verbalisation of this class negation is not supported yet.}\text{’}$
   18. end if
   19. end if
20. return RESULT
**Algorithm 4 (Negation)** Verbalisation of negation in an axiom, as disjointness or negated object property (i.e., axioms of type $C \subseteq \neg D$ and $C \subseteq \neg \exists R. D$).

**Require:** $C$ set of classes, language $L$ with $\subseteq$ for subsumption and $\neg$ for negation; variables:
- $A$ axiom, $NC_i$ noun class, $c_1, c_2 \in C$, $a_1$ term, $a_2$ letter and $n, p$ are concords, $v$ verb stem;
- functions: $checkNegation(A)$, $getNSC(NC_i)$, $getPNC(NC_i)$.

**Require:** $checkNegation(A) == true$
1: if negation directly preceded by $\subseteq$ and directly followed by $c_2$ then
2: \hspace{1em} $NC'_1 \leftarrow$ lookup plural nounclass of $NC_1$
3: \hspace{2em} $c'_1 \leftarrow$ pluralise($c_1, NC'_1$) \hspace{1em} \{ call algorithm pluralise to generate a plural from $o$ \}
4: \hspace{2em} $a_1 \leftarrow$ lookup quantitative concord for $NC'_1$ \hspace{1em} \{ from quantitative concord (QC(all)) list \}
5: \hspace{2em} $n \leftarrow$ getNSC($NC'_1$) \hspace{2em} \{ get negative subject concord for $c'_1$ \}
6: \hspace{2em} $p \leftarrow$ getPNC($NC_2$) \hspace{2em} \{ get pronominal for $c_2$ \}
7: \hspace{2em} RESULT $\leftarrow$ ‘$a_1 \hspace{0.1em} c'_1 \hspace{0.1em} np \hspace{0.1em} c_2.$’ \hspace{2em} \{ verbalise the disjointness ($c_2$ is QC(all)) \}
8: else if negation in front of OP then
9: \hspace{2em} $v' \leftarrow$ remove final vowel of $v$ \hspace{2em} \{ i.e., obtain the (possibly extended) verb root \}
10: \hspace{2em} $n \leftarrow$ getNSC($NC'_1$) \hspace{2em} \{ get negative subject concord for $c'_1$ \}
11: \hspace{2em} if $v' \in \{a, e, i, o, u, \}$ then
12: \hspace{3em} $negv \leftarrow$ phonoCondNegSc($v'$, $n$)
13: \hspace{2em} else
14: \hspace{3em} $negv \leftarrow n + v'$
15: \hspace{2em} end if
16: \hspace{2em} RESULT $\leftarrow$ ‘$a_1 \hspace{0.1em} c'_1 \hspace{0.1em} negv \hspace{0.1em} c_2 \hspace{0.1em} r2q2dwa.$’ \hspace{2em} \{ verbalise the axiom \}
17: else
18: \hspace{2em} RESULT $\leftarrow$ ‘verticalisation of this class negation is not supported yet.’ \hspace{2em} \{ negation in front of $c_2$ and $A$ contains an OP \}
19: end if
20: return RESULT
Algorithm 4 (Negation) Verbalisation of negation in an axiom, as disjointness or negated object property (i.e., axioms of type $C \subseteq \neg D$ and $C \subseteq \neg \exists R.D$).

Require: $C$ set of classes, language $\mathcal{L}$ with $\sqsubseteq$ for subsumption and $\neg$ for negation; variables: $A$ axiom, $NC_i$ noun class, $c_1, c_2 \in C$, $a_1$ term, $a_2$ letter and $n, p$ are concords, $v$ verb stem; functions: checkNegation$(A)$, getNSC$(NC_i)$, getPNC$(NC_i)$.

Require: checkNegation$(A)$ == true
1: if negation directly preceded by $\sqsubseteq$ and directly followed by $c_2$ then
2: $NC'_1 \leftarrow$ lookup plural nounclass of $NC_1$
3: $c'_1 \leftarrow$ pluralise$(c_1, NC'_1)$ {call algorithm pluralise to generate a plural from o}
4: $a_1 \leftarrow$ lookup quantitative concord for $NC'_1$
5: $n \leftarrow$ getNSC$(NC'_1)$ {get negative subject concord for $c'_1$}
6: $p \leftarrow$ getPNC$(NC_2)$ {get pronomial for $c_2$}
7: RESULT $\leftarrow$ ‘$a_1$ $c'_1$ np $c_2$. ‘ {verbalise the disjointness ($c_1$ is QC(all))}
8: else if negation in front of OP then
9: $v' \leftarrow$ remove final vowel of $v$
10: $n \leftarrow$ getNSC$(NC'_1)$ {i.e., obtain the (possibly extended) verb root}
11: if $v' \in \{a, e, i, o, u, \}$ then
12: negv $\leftarrow$ phonoCondNegSc($v'$, $n$) \[ -(aX | eX) \rightarrow \text{negscyword} \]
13: else
14: negv $\leftarrow$ $n + v'$
15: end if
16: RESULT $\leftarrow$ ‘$a_1$ $c'_1$ negvi $c_2$ $r_2q_2$ dwa.’ {verbalise the axiom}
17: else
18: RESULT $\leftarrow$ ‘*verbalisation of this class negation is not supported yet.*’
19: end if
20: return RESULT
Question ‘templates’/patterns (the CNL)

- Sentence patterns: `<noun> <verb>` or `<noun> <verb> <noun>`
- `<noun>` constructed from prefix[SG/PL] + stem
- `<verb>` constructed from [Negative]Subject Concord + VerbRoot + [Negative]FinalVowel
- Takes into account phonological conditioning
1. Q: \(<\text{prefixSG+stem}> <\text{PLSC+VerbRoot+FV}>\)
   A: \(<\text{prefixPL+stem}> <\text{PLSC+VerbRoot+FV}>\)
   Q: \(<\text{prefixSG+stem}> <\text{PLSC+VerbRoot+FV}> <\text{prefixSG+stem}>\)
   A: \(<\text{prefixPL+stem}> <\text{PLSC+VerbRoot+FV}> <\text{prefixSG+stem}>\)

2. Q: \(<\text{prefixPL+stem}> <\text{SGSC+VerbRoot+FV}>\)
   A: \(<\text{prefixPL+stem}> <\text{PLSC+VerbRoot+FV}>\)

3. Q: \(<\text{prefixSG+stem}> <\text{SGSC+VerbRoot+FV}>\)
   A: \(<\text{prefixPL+stem}> <\text{PLSC+VerbRoot+FV}>\)
   Q: \(<\text{prefixSG+stem}> <\text{SGSC+VerbRoot+FV}> <\text{prefixSG+stem}>\)
   A: \(<\text{prefixPL+stem}> <\text{PLSC+VerbRoot+FV}> <\text{prefixPL+stem}>\)

4. Q: \(<\text{PLSC+VerbRoot+FV}>\)
   A: \(<\text{PLNEGSC+VerbRoot+NEGFW}>\)
Question ‘templates’/patterns (the CNL), combining components

- May mix and match the ‘slots’ (not tested); e.g.:

  Q: <prefixSG+stem> <SGSC+VerbRoot+FV> <prefixSG+stem>
  A: <prefixPL+stem> <PLNEGSC+VerbRoot+NEGFV> <prefixPL+stem>

- Example:

  *umfowethu uwasha inkomishi*
  ‘(my) brother washes the cup’

  *abafowethu abawashi izinkomishi*
  ‘(my) brothers do not wash the cups’

- The current system can generate 39501 question sentences and compute their answers (and 60 scrabble general common conversational sentences)
Outline

1. Motivation
2. Design
3. Evaluation and discussion
4. Conclusions
Preliminary evaluation

- Evaluation with an oracle (1 person who speaks isiZulu)
- Data analysis with some input from isiZulu linguist
Preliminary evaluation

- Evaluation with an oracle (1 person who speaks isiZulu)
- Data analysis with some input from isiZulu linguist
- Meaningfulness of the sentences and the grammatical correctness
- 30 sentences generated (15 singular, 15 plural), covering each type of template
- weigh each sentence equally, 1 or 0, calculate percentage
- space for comments on each sentence
Preliminary evaluation – Results

- Two-words sentences: 100% semantically meaningful and 96% grammatically correct (ticking a box omission)
- Three-word sentences: 63% semantically meaningful and 58% grammatically correct (at a first pass)
  - Words in the corpus and the ported pluraliser and conjugator
Preliminary evaluation – Results

- Two-words sentences: 100% semantically meaningful and 96% grammatically correct (ticking a box omission)
- Three-word sentences: 63% semantically meaningful and 58% grammatically correct (at a first pass)
  - Words in the corpus and the ported pluraliser and conjugator

**pluraliser:** e.g., *ushukela* ‘sugar’ has no plural (correct in [Byamugisha et al. (2018)])
- pluraliser coverage (aba- plural with abe- exception)
  - These issues affected 5 sentences
Preliminary evaluation – Results

- Two-words sentences: 100% semantically meaningful and 96% grammatically correct (ticking a box omission)
- Three-word sentences: 63% semantically meaningful and 58% grammatically correct (at a first pass)
  - Words in the corpus and the ported pluraliser and conjugator

pluraliser: e.g., *ushukela* ‘sugar’ has no plural (correct in [Byamugisha et al. (2018)])); pluraliser coverage (*aba*- plural with *abe*- exception)
  - These issues affected 5 sentences

corpus: *debatables*; e.g. *-sheka*: exists as is? (defecate, to be scared, or to commit something), or acceptable (or not) colloquial contraction of *shiyeka* ‘stay behind’; *udadewenu* or *ude wenu* ‘your sister’
  - These issues affected 7 sentences; not CALL’s problem
Preliminary evaluation – Results

- Two-words sentences: 100% semantically meaningful and 96% grammatically correct (ticking a box omission)
- Three-word sentences: 63% semantically meaningful and 58% grammatically correct (at a first pass)
  - Words in the corpus and the ported pluraliser and conjugator
  - pluraliser: e.g., *ushukela* ‘sugar’ has no plural (correct in [Byamugisha et al.(2018)]); pluraliser coverage (*aba*- plural with *abe*- exception)
  - These issues affected 5 sentences

- corpus: debatables; e.g. -*sheka*: exists as is? (defecate, to be scared, or to commit something), or acceptable (or not) colloquial contraction of *shiyeka* ‘stay behind’; *udadewenu* or *udade wenu* ‘your sister’
  - These issues affected 7 sentences; not CALL’s problem

- Thus, the CNL templates function as intended, the underlying algorithms perform mostly well (and updated), and the word chaining process also works well
Discussion

- NV and NVN sentence structure may look simple
- e.g. in English negation is a simple: ‘does not’ or ‘do not’ regardless who or what the subject is and regardless the morphology of the verb
- yet, in isiZulu: 12 singular NCs + 9 plural NC combinations with singulars + 6 personal pronouns = 27 negative SCs to consider, and then a set of phonological conditioning rules
- Or: range of templates may seem small, but the variability of what can possibly be slotted in is much higher
Discussion

- Our CALL system provides many more exercises than the paper-based versions
- Basic vocabulary used in a versatile way
- Addresses also things like the “practice effect”
- Conducted preliminary experiments with assigning difficulty levels to the exercises (integrated in the system presented) that aims to contribute to assessing the learner’s level and progress
Conclusions

- New CALL exercises, CNL-based
- Small new corpus
- Algorithms to compute the answers that adhere to the specified answer templates
- Modular approach
- 100% semantic accuracy for two-word isiZulu sentences; room for improvement for three-word sentences
- Exercise extensions include the object concord and past tense, a larger corpus, and more comprehensive testing
References I

Joan Byamugisha, C. Maria Keet, and Langa Khumalo.
Pluralising nouns in isiZulu and similar languages.

C. Gardent and L. Perez-Beltrachini.
Using FB-LTAG Derivation Trees to Generate Transformation-Based Grammar Exercises.
Sep 2012, Paris, France.

C. M. Keet and L. Khumalo.
Toward a knowledge-to-text controlled natural language of isiZulu.

C. Maria Keet and Langa Khumalo.
Basics for a grammar engine to verbalize logical theories in isiZulu.
August 18-20, 2014, Prague, Czech Republic.

M. Prabitha.
Computer assisted language learning: Benefits and barriers.
Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach.
Ukwabelana – an open-source morphological zulu corpus.
Thank you!

Questions?

More details are available at http://www.meteck.org/files/geni/