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

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 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
2.boy	**2**.small	**2.SUBJ**-FUT-**10.OBJ**-buy			**10**.book	REL-**10**.big

'The little boys will buy the big books'

Noun classes (simplified)

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

Related works

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

Related works

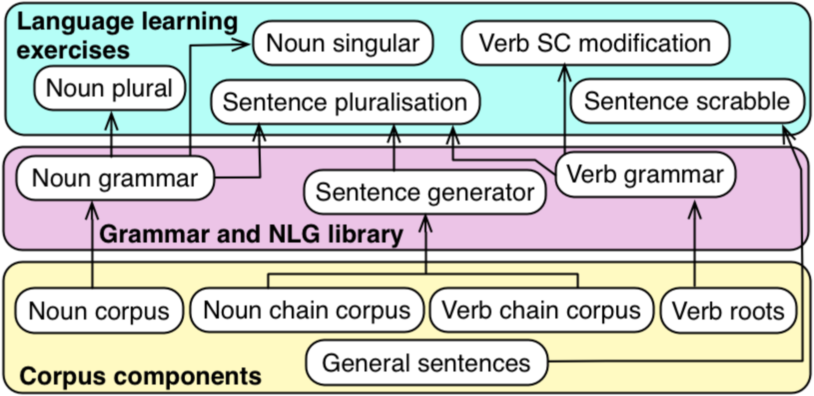
- Use grammar banks, like transformation-based grammar exercises (e.g., [Gardent and Perez-Beltrachini(2011)])
  - Very limited documented isiZulu grammar
- Corpus-based, POS tagged (e.g., [Sinclair(2004)]);
  - Outdated and out-of domain text [Spiegler et al.(2010)]; very limited other POS tagged text

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)

Outline

Architecture of the back-end 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

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/`

Small example of reuse

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: $checkNegation(A)$, $getNSC(NC_i)$, $getPNC(NC_i)$.

Require: $checkNegation(A) == true$

if negation directly preceded by $\sqsubseteq$ **and** directly followed by c_2 **then**
  $NC'_1 \leftarrow$ lookup plural nounclass of NC_1 {from known list}
  $c'_1 \leftarrow pluralise(c_1, NC'_1)$ {call algorithm $pluralise$ to generate a plural from o}
  $a_1 \leftarrow$ lookup quantitative concord for NC'_1 {from quantitative concord (QC(all)) list}
  $n \leftarrow getNSC(NC'_1)$ {get negative subject concord for c'_1}
  $p \leftarrow getPNC(NC_2)$ {get pronomial for c_2}
  RESULT $\leftarrow$ 'a_1 c'_1 np c_2.' {verbalise the disjointness (a_1 is QC(all))}
else if negation in front of OP **then**
  $v' \leftarrow$ remove final vowel of v {i.e., obtain the (possibly extended) verb root}
  $n \leftarrow getNSC(NC'_1)$ {get negative subject concord for c'_1}
  if $v' \in \{a, e, i, o, u,\}$ **then**
    $negv \leftarrow phonoCondNegSc(v', n)$
  else
    $negv \leftarrow n + v'$
  end if
  RESULT $\leftarrow$ 'a_1 c'_1 $negv$i c_2 $r_2q_2$dwa.' {verbalise the axiom}
else {negation in front of c_2 **and** A contains an OP}
  RESULT $\leftarrow$ '*verbalisation of this class negation is not supported yet.*'
end if
return RESULT

Small example of reuse

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: $checkNegation(A)$, $getNSC(NC_i)$, $getPNC(NC_i)$.
Require: $checkNegation(A) == true$

```
if negation directly preceded by ⊑ and directly followed by c2 then
    NC'1 ← lookup plural nounclass of NC1                    {from known list}
    c'1 ← pluralise(c1, NC'1)       {call algorithm pluralise to generate a plural from o}
    a1 ← lookup quantitative concord for NC'1   {from quantitative concord (QC(all)) list}
    n ← getNSC(NC'1)                        {get negative subject concord for c'1}
    p ← getPNC(NC2)                                      {get pronomial for c2}
    RESULT ← 'a1 c'1 np c2.'          {verbalise the disjointness (a1 is QC(all))}
else if negation in front of OP then
    v' ← remove final vowel of v    {i.e., obtain the (possibly extended) verb root}
    n ← getNSC(NC'1)                        {get negative subject concord for c'1}
    if v' ∈ {a, e, i, o, u, } then
        negv ← phonoCondNegSc(v', n)
    else
        negv ← n + v'
    end if
    RESULT ← 'a1 c'1 negvi c2 r2q2dwa.'                     {verbalise the axiom}
else                                 {negation in front of c2 and A contains an OP}
    RESULT ← 'verbalisation of this class negation is not supported yet.'
end if
return RESULT
```

Small example of reuse

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: $checkNegation(A)$, $getNSC(NC_i)$, $getPNC(NC_i)$.

Require: $checkNegation(A) == true$

if negation directly preceded by $\sqsubseteq$ **and** directly followed by c_2 **then**
- $NC'_1 \leftarrow$ lookup plural nounclass of NC_1 {from known list}
- $c'_1 \leftarrow pluralise(c_1, NC'_1)$ {call algorithm $pluralise$ to generate a plural from o}
- $a_1 \leftarrow$ lookup quantitative concord for NC'_1 {from quantitative concord (QC(all)) list}
- $n \leftarrow getNSC(NC'_1)$ {get negative subject concord for c'_1}
- $p \leftarrow getPNC(NC_2)$ {get pronomial for c_2}
- RESULT $\leftarrow$ 'a_1 c'_1 np c_2.' {verbalise the disjointness (a_1 is QC(all))}

else if negation in front of OP **then**
- $v' \leftarrow$ remove final vowel of v {i.e., obtain the (possibly extended) verb root}
- $n \leftarrow getNSC(NC'_1)$ {get negative subject concord for c'_1}
- **if** $v' \in \{a, e, i, o, u,\}$ **then**
  - $negv \leftarrow phonoCondNegSc(v', n)$
- **else**
  - $negv \leftarrow n + v'$
- **end if**
- RESULT $\leftarrow$ 'a_1 c'_1 $negv$i c_2 $r_2 q_2$dwa.' {verbalise the axiom}

else {negation in front of c_2 **and** A contains an OP}
- RESULT $\leftarrow$ '*verbalisation of this class negation is not supported yet.*'

end if

return RESULT

– (aX | eX) $\rightarrow$ *negscyword*

– (iX | oX | uX) $\rightarrow$ *negscngword*

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

Question ‘templates’/patterns (the CNL), selection

1. Q: <*prefixSG*+stem> <*PLSC*+VerbRoot+FV>
   A: <prefixPL+stem> <PLSC+VerbRoot+FV>
   Q: <*prefixSG*+stem> <*PLSC*+VerbRoot+FV> <prefixSG+stem>
   A: <prefixPL+stem> <PLSC+VerbRoot+FV> <prefixSG+stem>

2. Q: <*prefixPL*+stem> <*SGSC*+VerbRoot+FV>
   A: <prefixPL+stem> <PLSC+VerbRoot+FV>

3. Q: <prefixSG+stem> <SGSC+VerbRoot+FV>
   A: <prefixPL+stem> <PLSC+VerbRoot+FV>
   Q: <prefixSG+stem> <SGSC+VerbRoot+FV> <prefixSG+stem>
   A: <prefixPL+stem> <PLSC+VerbRoot+FV> <prefixPL+stem>

4. Q: <PLSC+VerbRoot+FV>
   A: <PLNEGSC+VerbRoot+NEGFV>

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

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 *udade 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 looks 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

Outline

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.
In A. Gelbukh, editor, *Proceedings of CICLing'16*, volume 9623 of *LNCS*, pages 271–283. Springer, 2018.

C. Gardent and L. Perez-Beltrachini.
Using FB-LTAG Derivation Trees to Generate Transformation-Based Grammar Exercices.
In *Proc. of TAG+11*, pages 117–125. ACL, 2011.
Sep 2012, Paris, France.

C. M. Keet and L. Khumalo.
Toward a knowledge-to-text controlled natural language of isiZulu.
Language Resources and Evaluation, 51(1):131–157, 2017.
doi: 10.1007/s10579-016-9340-0.

C. Maria Keet and Langa Khumalo.
Basics for a grammar engine to verbalize logical theories in isiZulu.
In A. Bikakis et al., editors, *Proceedings of the 8th International Web Rule Symposium (RuleML'14)*, volume 8620 of *LNCS*, pages 216–225. Springer, 2014.
August 18-20, 2014, Prague, Czech Republic.

M. Prabitha.
Computer assisted language learning: Benefits and barriers.
Journal of Literature, Culture and Media Studies, 2:59–71, 2010.

References II

Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach.
Ukwabelana – an open-source morphological zulu corpus.
In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, pages 1020–1028. Association for Computational Linguistics, 2010.
Beijing.

Thank you!

Questions?

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