Natural Language Generation for Agglutinating African Languages – A brief overview

C. Maria Keet

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

SADiLaR Digital Humanities Colloquium, online, 18 May 2022

Outline

1. Motivation
   - Context
   - Notes on NCB languages
2. Rule-based NLG
   - What is CNL, NLG?
   - Generating basic sentences in isiZulu
   - Extending basic sentences
3. On broadening and generalising results
   - Other languages
   - Use of the algorithms in applications
4. Summary

Outline

Outline

Motivation

- >1 billion people in Africa, most do or can speak a language other than English or French
  - e.g., South Africa: IsiZulu and isiXhosa most widely spoken languages, by first language speakers
  - 23% or about 11 million people isiZulu, 8 million (isiXhosa)

Motivation

- >1 billion people in Africa, most do or can speak a language other than English or French
  - e.g., South Africa: IsiZulu and isiXhosa most widely spoken languages, by first language speakers
  - 23% or about 11 million people isiZulu, 8 million (isiXhosa)
- People use computers for work, social media...
  - Doing business, government services provision, etc in one's own language, beyond English and French
  - (The "untapped billion", in FAANG's terminology)
- ... but there is very limited ICT support

Motivation

- NLP tools also for African languages proper (not just MT through English)
- Requires tools with African languages in at least the interface, not just some ‘pretty pictures and icons’
- Need to transform structured data and structured knowledge into text
- Structured input is represented in, a.o.: XML, RDF, OWL, SQL, JSON, spreadsheets, csv files

Structured input – examples

OWL snippet:


```xml
-<!--
   http://www.meteck.org/teaching/OEbook/ontologies/AfricanWildlifeOntology1.owl#CarnivorousPlant
  -->
-<owl:Class rdf:about="http://www.meteck.org/teaching/OEbook/ontologies/AfricanWildlifeOntology1.owl#CarnivorousPlant">
   <rdfs:subClassOf rdf:resource="http://www.meteck.org/teaching/OEbook/ontologies/AfricanWildlifeOntology1.owl#plant"/>
  -<rdfs:subClassOf>
    -<owl:Restriction>
       <owl:onProperty rdf:resource="http://www.meteck.org/teaching/OEbook/ontologies/AfricanWildlifeOntology1.owl#eats"/>
       <owl:someValuesFrom rdf:resource="http://www.meteck.org/teaching/OEbook/ontologies/AfricanWildlifeOntology1.owl#animal"/>
     </owl:Restriction>
   </rdfs:subClassOf>
 </owl:Class>
```


JSON:


```
▼ relationshipTypes:
  ▼ 0:
      name:              "Depencency"
    ▼ participants:
      ▼ 0:
          name:            "Employee"
          role:            "provides_for"
          participation:   "strong"
          min:             "0"
          max:             "N"
      ▼ 1:
          name:            "Dependent"
          role:            "supported"
          participation:   "weak"
          min:             "1"
          max:             "1"
```


XML:


```xml
-<CATALOG>
  -<PLANT>
     <COMMON>Bloodroot</COMMON>
     <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
     <ZONE>4</ZONE>
     <LIGHT>Mostly Shady</LIGHT>
     <PRICE>$2.44</PRICE>
     <AVAILABILITY>031599</AVAILABILITY>
   </PLANT>
  -<PLANT>
     <COMMON>Columbine</COMMON>
```

Structured sentences – examples for knowledge-to-text

- Electronic health records and patient discharge notes generation

- Getting the relevant business logic into your app

- Querying the data with conceptual queries in OBDA

- And many other areas; e.g., question generation, intelligent textbooks, automation of language learning exercises

Structured sentences – examples for knowledge-to-text

- Electronic health records and patient discharge notes generation
  - e.g., SNOMED CT, OpenMRS localisation
  - "The patient has as symptom fever and dizziness"
  - "The patient must drink water when taking the pills"
    "If the patient takes the pills, then he must drink water"
- Getting the relevant business logic into your app
  - Requirements engineering, competency questions
  - "Which animals eat impalas?"
- Querying the data with conceptual queries in OBDA
  - "Show me all employees who are not working on a project"
- And many other areas; e.g., question generation, intelligent textbooks, automation of language learning exercises

Structured sentences – examples for knowledge-to-text

- Electronic health records and patient discharge notes generation
  - e.g., SNOMED CT, OpenMRS localisation
  - "The patient has as symptom fever **and** dizziness"
  - "The patient must drink water **when** taking the pills"
    "**If** the patient takes the pills, **then** he must drink water"
- Getting the relevant business logic into your app
  - Requirements engineering, competency questions
  - "**Which** animals eat impalas?"
- Querying the data with conceptual queries in OBDA
  - "Show me **all** employees who are **not** working on a project"
- And many other areas; e.g., question generation, intelligent textbooks, automation of language learning exercises

This talk

- Zooming in on:
  - Controlled Natural Languages
  - Natural Language Generation
- Sample use cases of the techniques
  - Language learning: automated question generation and marking
  - Financial inclusion (writing out numbers, toward TTS)

Outline

Basics

1500-2000 African languages (6 main groups) spoken by 1.2 billion people

Core characteristics relevant for computation (1/2)

- System of noun classes
  - Each noun is classified into a noun class
  - Meinhof identified 23 noun classes; not all of them used, varies by language; some refinements
  - Singular and plural pairings (with imprecision and underspecification)
  - There's semantics to the NCs (e.g., NC1 for humans, NC9 for animals, NC15 infinitive nouns); less important for computation

NC	AU	PRE	Stem (example)	Meaning	Example (isiZulu)	
1 2	u- a-	m(u)- ba-	-fana -fana	humans and other animates	umfana abafana	boy boys
1a 2a	u- o-	- -	-baba -baba	kinship terms and proper names	ubaba obaba	father fathers
3a (2a)	u- o-	- -	-shizi -shizi	nonhuman	ushizi oshizi	cheese cheeses
3 4	u- i-	m(u)- mi-	-fula -fula	trees, plants, non-paired body parts	umfula imifula	river rivers
5 6	i- a-	(li)- ma-	-gama -gama	fruits, paired body parts, and natural phenomena	igama amagama	name names
7 8	i- i-	si- zi-	-hlalo -hlalo	inanimates and manner/ style	isihlalo izihlalo	chair chairs
9a (6)	i- a-	- ma-	-rabha -rabha	nonhuman	irabha amarabha	rubber rubbers
9 10	i(n)- i-	- zi(n)-	-ja -ja	animals	inja izinja	dog dogs
11 (10)	u- i-	(lu)- zi(n)-	-thi -thi	inanimates and long thin objects	uthi izinthi	stick sticks
14 15	u- u-	bu- ku-	-hle -cula	abstract nouns infinitives	ubuhle ukucula	beauty to sing
17		ku-		locatives, remote/ general		locative

Core characteristics relevant for computation (2/2)

- Many of the languages are *agglutinating*
  - i.e., what are separate words in, say, English are 'components' of a word

Ex: `titukakimureeterahoganu` (Runyankore, Uganda)
'We have never ever brought it to him'

Core characteristics relevant for computation (2/2)

- Many of the languages are *agglutinating*
  - i.e., what are separate words in, say, English are 'components' of a word

Ex: `titukakimureeterahoganu` (Runyankore, Uganda)
'We have never ever brought it to him'
`ti tu ka ki mu reet er a ho ga nu`
neg-(NC2 SC)-RM-(NC7 SC)-(NC1 SC)-VR-App-FV-Loc-Emp-Dec

Core characteristics relevant for computation (2/2)

- Many of the languages are *agglutinating*
  - i.e., what are separate words in, say, English are ‘components’ of a word

Ex: `titukakimureeterahoganu` (Runyankore, Uganda)
‘We have never ever brought it to him’
`ti tu ka ki mu reet er a ho ga nu`
neg-(NC2 SC)-RM-(NC7 SC)-(NC1 SC)-VR-App-FV-Loc-Emp-Dec

- System of concordial agreement (more about that soon)

Illustrative examples of some consquences (isiZulu)

- 'and', enumerative: *na-*, phonologically conditioned

  Ex: milk and butter: *ubisi <u>ne</u>bhotela* (-a+i-=-e-)

  Ex: butter and milk: *ibhotela <u>no</u>bisi* (-a+u-=-o-)

Illustrative examples of some consquences (isiZulu)

- 'and', enumerative: *na-*, phonologically conditioned
  Ex: milk and butter: *ubisi nebhotela* (-a+i-=-e-)
  Ex: butter and milk: *ibhotela nobisi* (-a+u-=-o-)
- copulative (to be): depends on first letter of noun: *ng-* for a-, o-, u-, else *y-*
  Ex: is a dog: *yinja*
  Ex: is a grandmother: *ngugogo*
- 'is not a': combine NEG SC with PRON, both depend on noun class
  Ex: an animal is not a plant: *isilwane asiwona umuthi*
  Ex: a plant is not an animal: *umuthi awusona isilwane*
- Other verbs: concordial agreement ($\sim$ conjugation) based on noun class
  Ex: The human eats: *umuntu udla*
  Ex: The dog eats: *inja idla*

Concordial agreement—example (isiZulu, South Africa)

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'

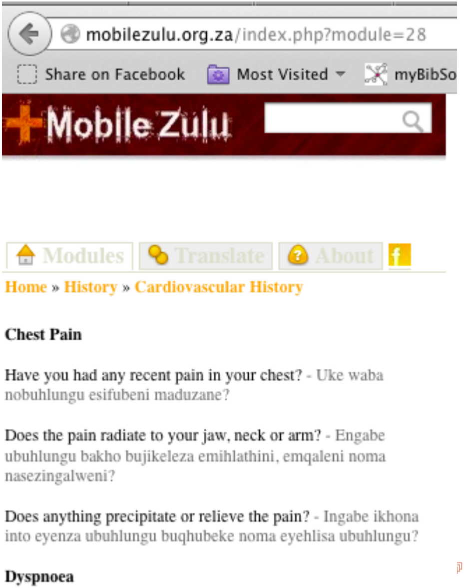
Outline

Outline

Short answer

- **C**controlled **N**atural **L**anguage: constrain the grammar or vocabulary (or both) of a natural language
- **N**atural **L**anguage **G**eneration: generate natural language text from structured data, information, or knowledge

Ex: S. Moolla's mobile healthcare app with **canned text**



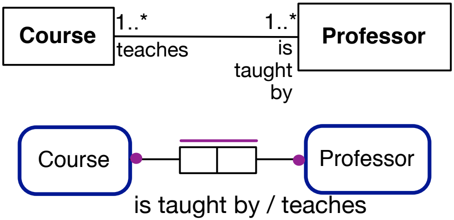
Ex: Avalanche bulletins with **canned segments** [Winkler et al.(2014)]

Segment 1	Segment 2	Segment 3	Segment 4	Segment 5
die Lawinen	**können**			gross werden.
nasse Lawinen		auch	oft	weit vorstossen.
diese		**{on_steep} Sonnenhängen**	weiterhin	bis in die aperen Täler vorstossen.
		in diesen Gebieten		**bis in tiefe Lagen vorstossen.**

Segment 3a	Segment 1	Segment 2	Segment 3b	Segment 4	Segment 5
	the avalanches	**can**			reach large size.
	wet avalanches		also	in many cases	reach a long way.
{on_steep} sunny slopes	they			as before	reach the bare valleys.
in these regions					**reach low altitudes.**

Fig. 2. Schema of a phrase in the source language German (above). {on_steep} mark a sub-segment with several further options. In this example, [blank] is one of the options in the third and fourth segment. In English, the order of the segments is different and segment 3 is split.

Ex: Business rules and conceptual data models with *static* **templates**



Each Course is taught by **at least one** Professor
Each Professor teaches **at least one** Course

Ex.: Mixing grammar with templates

- Idea: store the words in their base form with POS tag, specify in the 'template' what needs to be done with it, use a realisation engine to finalise the sentence
- e.g., yes/no pronomial or gender as variables to set

Ex.: Mixing grammar with templates

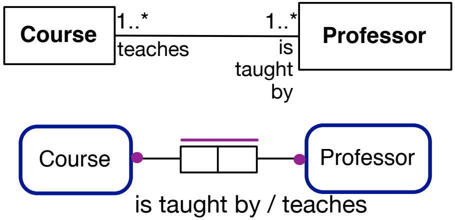
- Idea: store the words in their base form with POS tag, specify in the 'template' what needs to be done with it, use a realisation engine to finalise the sentence
- e.g., yes/no pronomial or gender as variables to set
- Same stems or words and core structure of the grammar-infused template, geenrate different sentences; e.g.:
  John eats an apple
  He eats an apple
  He eats it
  John eats it
  ...

NLG, principal approaches to generate the text

- ~~Canned text, with complete sentences (CNLs only)~~
- Canned segments to make a sentence (CNL mostly, not NLG)
- Templates (different types)
  - Mainly for English but also other languages
  - Hand-crafted ('old' approach) or ML/neural-based ('new')
- Grammar engines, such as [Kuhn(2013)], Grammatical Framework (`http://www.grammaticalframework.org/`), SimpleNLG [Gatt and Reiter(2009)]

See also: [Mahlaza and Keet(2020)]

Business rules/conceptual data models and logic reconstruction



BR: **Each** Course is taught by **at least one** Professor

FOL: $\forall x\ (\text{Course}(x) \rightarrow \exists y\ (\text{is\_taught\_by}(x, y) \land \text{Professor}(y)))$

DL: $\text{Course} \sqsubseteq \exists\ \text{is\_taught\_by}.\text{Professor}$

- (i.e., a mandatory constraint / existential quantification)

Example of static templates

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory] Cada</Text>
 <Object index="0"/>
 <Text>debe</Text>
 <Role index="0"/>
 <Text>al menos un(a)</Text>
 <Object index="1"/>
</Constraint>
```

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory]  Each</Text>
 <Object index="0"/>
 <Text>must</Text>
 <Role index="0"/>
 <Text>at least one</Text>
 <Object index="1"/>
</Constraint>
```

for a large fragment of ORM, and 11 languages [Jarrar et al.(2006)]

Example of static templates

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory] Cada</Text>
 <Object index="0"/>
 <Text>debe</Text>
 <Role index="0"/>
 <Text>al menos un(a)</Text>
 <Object index="1"/>
</Constraint>
```

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory]  Each</Text>
 <Object index="0"/>
 <Text>must</Text>
 <Role index="0"/>
 <Text>at least one</Text>
 <Object index="1"/>
</Constraint>
```

for a large fragment of ORM, and 11 languages [Jarrar et al.(2006)]

Example of static templates

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory] Cada</Text>
 <Object index="0"/>
 <Text>debe</Text>
 <Role index="0"/>
 <Text>al menos un(a)</Text>
 <Object index="1"/>
</Constraint>
```

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory]  Each</Text>
 <Object index="0"/>
 <Text>must</Text>
 <Role index="0"/>
 <Text>at least one</Text>
 <Object index="1"/>
</Constraint>
```

for a large fragment of ORM, and 11 languages [Jarrar et al.(2006)]

Example of static templates

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory] Cada</Text>
 <Object index="0"/>
 <Text>debe</Text>
 <Role index="0"/>
 <Text>al menos un(a)</Text>
 <Object index="1"/>
</Constraint>
```

```
<Constraint xsi:type="Mandatory">
 <Text> -[Mandatory]  Each</Text>
 <Object index="0"/>
 <Text>must</Text>
 <Role index="0"/>
 <Text>at least one</Text>
 <Object index="1"/>
</Constraint>
```

for a large fragment of ORM, and 11 languages [Jarrar et al.(2006)]

John eats apples – fancier templates

```
((template clause)
  (act 'eat')
  (agent ((template noun-phrase)
    (np-type PROPER)
    (head 'John')
    (gender MASCULINE)
    (pronominal NO)))
  (object ((template noun-phrase)
    (head 'apple')
    (pronominal YES))))
```

John eats it

```
((template clause)
  (act 'eat')
  (agent ((template noun-phrase)
    (np-type PROPER)
    (head 'John')
    (gender FEMININE)
    (pronominal YES)))
  (object ((template noun-phrase)
    (head 'apple')
    (pronominal NO))))
```

She eats an apple

NL Grammars, illustration (1/2)

$$
\begin{aligned}
\textit{Sentence} &\longrightarrow \textit{NounPhrase} \mid \textit{VerbPhrase} \\
\textit{NounPhrase} &\longrightarrow \textit{Adjective} \mid \textit{NounPhrase} \\
\textit{NounPhrase} &\longrightarrow \textit{Noun} \\
\ldots & \\
\textit{Noun} &\longrightarrow \textit{car} \mid \textit{train} \\
\textit{Adjective} &\longrightarrow \textit{big} \mid \textit{broken} \\
\ldots &
\end{aligned}
$$

(and complexity of the grammar)

+ rules for verb tenses, pluralisation etc.

SimpleNLG tool [Gatt and Reiter(2009)] (2/2)

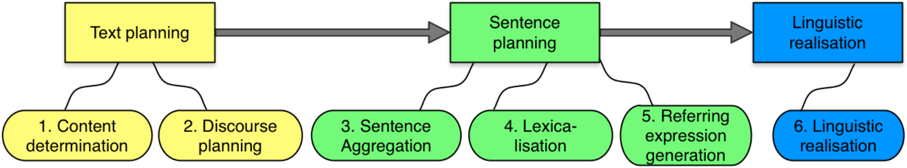
with grammars for EN, FR, ES, PT, NL, DE, and Galician

```
<Document>
  <child xsi:type="SPhraseSpec">
    <subj xsi:type="VPPhraseSpec" FORM="PRESENT_PARTICIPLE">
      <head cat="VERB">
        <base>refactor</base>
      </head>
    </subj>
    <vp xsi:type="VPPhraseSpec" TENSE="PRESENT" >
      <head cat="VERB">
        <base>be</base>
      </head>
      <compl xsi:type="VPPhraseSpec" FORM="PAST_PARTICIPLE">
        <head cat="VERB">
          <base>need</base>
        </head>
      </compl>
    </vp>
  </child>
</Document>
```

Generates: “Refactoring is needed”

https://github.com/simplenlg/simplenlg

The 'NLG pipeline'



1. What structured data/info/ knowledge do you want to put into NL sentences?
2. In what order should it be presented?

3. Which messages to put together into a sentence?
4. Which words and phrases will it use for each domain concept and relation?
5. Which words or phrases to select to identify domain entities?

6. Use grammar rules to produce syntactically, morphologically, and orthographically correct (and is also meaningful)

Outline

Question

- Can we use any of these approaches for agglutinating Niger-Congo B languages?

Question

- Can we use any of these approaches for agglutinating Niger-Congo B languages?
  - It depends... but mostly: no

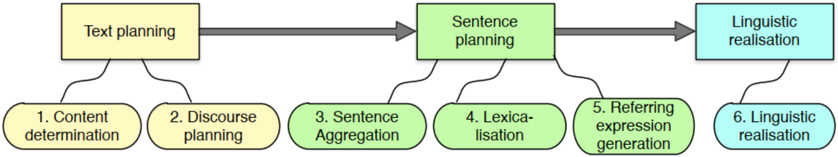
Question

- Can we use any of these approaches for agglutinating Niger-Congo B languages?
  - It depends... but mostly: no
- Tasks:
  - For structured input: use a practically useful language with tool support already (Semantic Web technologies)
  - Start with basics for a grammar engine (develop the new algorithms)
  - Pick an appealing sample domain (e.g., health)
  - Do it in a way so as to benefit both ICT and linguists

Question

- Can we use any of these approaches for agglutinating Niger-Congo B languages?
  - It depends... but mostly: no
- Tasks:
  - For structured input: use a practically useful language with tool support already (Semantic Web technologies)
  - Start with basics for a grammar engine (develop the new algorithms)
  - Pick an appealing sample domain (e.g., health)
  - Do it in a way so as to benefit both ICT and linguists
- First language to experiment with: isiZulu [Keet and Khumalo(2014b), Keet and Khumalo(2014a), Keet and Khumalo(2017a)]

Ontology verbalisation



1. What structured data/info/ knowledge do you want to put into NL sentences?
2. In what order should it be presented?

3. Which messages to put together into a sentence?
4. Which words and phrases will it use for each domain concept and relation?
5. Which words or phrases to select to identify domain entities?

6. Use grammar rules to produce syntac-tically, morphological-ly, and orthographical-ly correct (and is also meaningful)

The NLG 'pipeline'

Ontology verbalisation

1. The (OWL) ontology
2. Your choice (e.g., first all classes and class expressions in the TBox, then the object properties, etc.)

3. Aim: sentence for each axiom
4. Use vocabulary of the ontology; Select term for each constructor in the language (Each/All, and, some/at least one)
5. Combine related small axiom, or to relate the sentences generated for a large axiom

6. Language-specific issues (e.g., singular/plural of the class in agreement with conjugation of the verb, 'a' and 'an' vs 'a(n)', etc.)

A logic foundation for isiZulu knowledge-to-text

- Roughly OWL 2 EL
- OWL 2 EL is a W3C-standardised profile of OWL 2
- Tools, ontologies in OWL 2 (notably SNOMED CT)

$\mathcal{ALC}$ syntax

- *Concepts* denoting entity types/classes/unary predicates/universals, including top $\top$ and bottom $\bot$;
- *Roles* denoting relationships/associations/n-ary predicates/properties;
- *Constructors*: and $\sqcap$, or $\sqcup$, and not $\neg$; quantifiers 'for all' $\forall$ and 'there exists' $\exists$
- *Complex concepts* using constructors: Let C and D be concept names, R a role name, then
  - $\neg C$, $C \sqcap D$, and $C \sqcup D$ are concepts, and
  - $\forall R.C$ and $\exists R.C$ are concepts
- *Individuals*
- e.g., $Lion \sqsubseteq \exists eats.Herbivore \sqcap \forall eats.Herbivore$

Universal Quantification

- Consider here only the universal quantification at the start of the concept inclusion axiom ('nominal head')
- 'all'/'each' uses *-onke*, prefixed with the oral prefix of the noun class of that first noun (OWL class/DL concept) on lhs of $\sqsubseteq$

(U1) `Boy` $\sqsubseteq$ ...

wonke umfana ... ('each boy...'; *u-* + *-onke*)

bonke abafana ... ('all boys...'; *ba-* + *-onke*)

(U2) `Phone` $\sqsubseteq$ ...

lonke ifoni ... ('each phone...'; *li-* + *-onke*)

onke amafoni ... ('all phones...'; *a-* + *-onke*)

NC	QC (all)		NEG SC	PRON	RC	QC_{dwa}	EC
	$QC_{oral+onke}$	QC_{nke}					
1	u-onke → wonke	wo-	aka-	yena	o-	ye-	mu-
2	ba-onke → bonke	bo-	aba-	bona	aba-	bo-	ba-
1a	u-onke → wonke	wo-	aka-	yena	o-	ye-	mu-
2a	ba-onke → bonke	bo-	aba-	bona	aba-	bo-	ba-
3a	u-onke → wonke	wo-	aka-	wona	o-	ye-	mu-
(2a)	ba-onke → bonke	bo-	aba-	bona	aba-	bo-	ba-
3	u-onke → wonke	wo-	awu-	wona	o-	wo-	mu-
4	i-onke → yonke	yo-	ayi-	yona	e-	yo-	mi-
5	li-onke → lonke	lo-	ali-	lona	eli-	lo-	li-
6	a-onke → onke	o-	awa-	wona	a-	wo-	ma-
7	si-onke → sonke	so-	asi-	sona	esi-	so-	si-
8	zi-onke → zonke	zo-	azi-	zona	ezi	zo-	zi-
9a	i-onke → yonke	yo-	ayi-	yona	e-	yo-	yi-
(6)	a-onke → onke	o-	awa-	wona	a-	wo-	ma-
9	i-onke → yonke	yo-	ayi-	yona	e-	yo-	yi-
10	zi-onke → zonke	zo-	azi-	zona	ezi-	zo-	zi-
11	lu-onke → lonke	lo-	alu-	lona	olu-	lo-	lu-
(10)	zi-onke → zonke	zo-	azi-	zona	ezi-	zo-	zi-
14	ba-onke → bonke	bo-	abu-	bona	obu-	bo-	bu-
15	ku-onke → konke	zo-	aku-	khona	oku-	zo-	ku-

NC	QC_{oral} QC (all) → -onke	QC_{nke}	NEG SC	PRON	RC	QC_{dwa}	EC
1	u-onke → wonke	wo-	aka-	yena	o-	ye-	mu-
2	ba-onke → bonke	bo-	aba-	bona	aba-	bo-	ba-
1a	u-onke → wonke	wo-	aka-	yena	o-	ye-	mu-
2a	ba-onke → bonke	bo-	aba-	bona	aba-	bo-	ba-
3a	u-onke → wonke	wo-	aka-	wona	o-	ye-	mu-
(2a)	ba-onke → bonke	bo-	aba-	bona	aba-	bo-	ba-
3	u-onke → wonke	wo-	awu-	wona	o-	wo-	mu-
4	i-onke → yonke	yo-	ayi-	yona	e-	yo-	mi-
5	li-onke → lonke	lo-	ali-	lona	eli-	lo-	li-
6	a-onke → onke	o-	awa-	wona	a-	wo-	ma-
7	si-onke → sonke	so-	asi-	sona	esi-	so-	si-
8	zi-onke → zonke	zo-	azi-	zona	ezi	zo-	zi-
9a	i-onke → yonke	yo-	ayi-	yona	e-	yo-	yi-
(6)	a-onke → onke	o-	awa-	wona	a-	wo-	ma-
9	i-onke → yonke	yo-	ayi-	yona	e-	yo-	yi-
10	zi-onke → zonke	zo-	azi-	zona	ezi-	zo-	zi-
11	lu-onke → lonke	lo-	alu-	lona	olu-	lo-	lu-
(10)	zi-onke → zonke	zo-	azi-	zona	ezi-	zo-	zi-
14	ba-onke → bonke	bo-	abu-	bona	obu-	bo-	bu-
15	ku-onke → konke	zo-	aku-	khona	oku-	zo-	ku-

Subsumption

- Two different ways of carving up the nouns to determine which rules apply: semantic and syntactic
- Need to choose between
  - singular and plural
  - with or without the universal quantification voiced
  - generic or determinate

(S1) `MedicinalHerb` $\sqsubseteq$ `Plant`

ikhambi <u>ngu</u>muthi ('medicinal herb <u>is</u> a plant')

amakhambi <u>yi</u>mithi ('medicinal herbs <u>are</u> plants')

<u>wonke</u> amakhambi <u>ngu</u>muthi ('<u>all</u> medicinal herbs <u>are</u> a plant')

(S2) (generic)

(S3) (determinate)

Possible subsumption patterns

a. N_1 <copulative *ng*/*y* depending on first letter of N_2>N_2.

b. <plural of N_1> <copulative *ng*/*y* depending on first letter of plural of N_2><plural of N_2>.

c. <All-concord for NC_x> + onke <plural of N_1, being of NC_x> <copulative *ng*/*y* depending on first letter of N_2>N_2.

Existential Quantification

(E1) Giraffe $\sqsubseteq$ $\exists$eats.Twig

yonke indlulamithi idla ihlamvana elilodwa ('each giraffe eats at least one twig')

zonke izindlulamithi zidla ihlamvana elilodwa ('all giraffes eat at least one twig')

a. <All-concord for NC_x> + onke <pl. N_1, is in NC_x> <conjugated verb> <N_2 of NC_y> <RC for NC_y><QC for NC_y> + dwa.

Example

- $\forall x\ (\text{Professor}(x) \rightarrow \exists y\ (\text{teaches}(x, y) \land \text{Course}(y)))$
- $\text{Professor} \sqsubseteq \exists\ \text{teaches}.\text{Course}$
- **Each** Professor teaches **at least one** Course

Example

- $\forall x\ (\text{uSolwazi}(x) \rightarrow \exists y\ (\text{-fundisa}(x,y) \land \text{Isifundo}(y)))$
- $\text{uSolwazi} \sqsubseteq \exists\ \text{-fundisa}.\text{Isifundo}$
- ?

$$\forall x\ (\text{uSolwazi}(x) \rightarrow \exists y\ (\text{-fundisa}(x, y) \land \text{Isifundo}(y)))$$

$$\text{uSolwazi} \sqsubseteq \exists\ \text{-fundisa}.\text{Isifundo}$$

$\forall x$ (uSolwazi(x) → ... (x, ...) ∧ ...

uSolwazi ⊑ ∃ -func...

look-up NC

pluralise

for-all

NC	AU	PRE
1	u-	m(u)-
2	a-	ba-
1a	u-	-
2a	o-	-
3a	u-	-
(2a)	o-	-
3	u-	m(u)-
4	i-	mi-
5	i-	(li)-
6	a-	ma-
7	i-	si-
8	i-	zi-
9a	i-	-
(6)	a-	ma-
9	i(n)-	-
10	i-	zi(n)-
11	u-	(lu)-
(10)	i-	zi(n)-
14	u-	bu-
15	u-	ku-
17		ku-

NC	QC (all) $QC_{oral+onke}$
1	u-onke → wonke
2	ba-onke → bonke
1a	u-onke → wonke
2a	ba-onke → bonke
3a	u-onke → wonke
(2a)	ba-onke → bonke
3	u-onke → wonke
4	i-onke → yonke
5	li-onke → lonke
6	a-onke → onke
7	si-onke → sonke
8	zi-onke → zonke
9a	i-onke → yonke
(6)	a-onke → onke
9	i-onke → yonke
10	zi-onke → zonke
11	lu-onke → lonke
(10)	zi-onke → zonke
14	ba-onke → bonke
15	ku-onke → konke

Bonke oSolwazi

$\forall x\ (\text{uSolwazi}(x) \rightarrow \exists y\ (\text{-fundisa}(x, y) \wedge \text{Isifundo}(y)))$

$\text{uSolwazi} \sqsubseteq \exists\ \text{-fundisa}.\text{Isifundo}$

reuse pluralised NC of subject

look-up SC of that NC

NC	SC
1	u-
2	ba-
1a	u-
2a	ba-
3a	u-
2a	ba-
3	u-
4	i-
5	li-
6	a-
7	si-
8	zi-
9a	i-
6	a-
9	i-
10	zi-
11	lu-
10	zi-
14	bu-
15	ku-
17	lu-

Bonke oSolwazi bafundisa

$$\forall x\ (\text{uSolwazi}(x) \rightarrow \exists y\ (\text{-fundisa}(x,y) \wedge \text{Isifundo}(y)))$$

$$\text{uSolwazi} \sqsubseteq \exists\ \text{-fundisa}.\text{Isifundo}$$

Bonke oSolwazi bafundisa Isifundo

$\forall x\ (\text{uSolwazi}(x) \rightarrow \exists y$ … $) \wedge$ … $))$

uSolwazi $\sqsubseteq \exists$ fundisa.Is…

look-up NC

get RC

get QC

add -dwa

NC	AU	PRE
1	u-	m(u)-
2	a-	ba-
1a	u-	-
2a	o-	-
3a	u-	-
(2a)	o-	-
3	u-	m(u)-
4	i-	mi-
5	i-	(li)-
6	a-	ma-
7	i-	si-
8	i-	zi-
9a	i-	-
(6)	a-	ma-
9	i(n)-	-
10	i-	zi(n)-
11	u-	(lu)-
(10)	i-	zi(n)-
14	u-	bu-
15	u-	ku-
17		ku-

RC	QC_{dwa}
o-	ye-
aba-	bo-
o-	ye-
aba-	bo-
o-	ye-
aba-	bo-
o-	wo-
e-	yo-
eli-	lo-
a-	wo-
esi-	so-
ezi	zo-
e-	yo-
a-	wo-
e-	yo-
ezi-	zo-
olu-	lo-
ezi-	zo-
obu-	bo-
oku-	zo-

Bonke oSolwazi bafundisa Isifundo esisodwa

Evaluation

- Typical way of evaluating: ask linguists and/or intended target group
- Questions depend on what you want to know; e.g.,
  - Does the text capture the semantics adequately?
  - Must it really be grammatically correct or is understandable also acceptable?
  - Compared against alternate representation (figures, tables) or human-authored text?

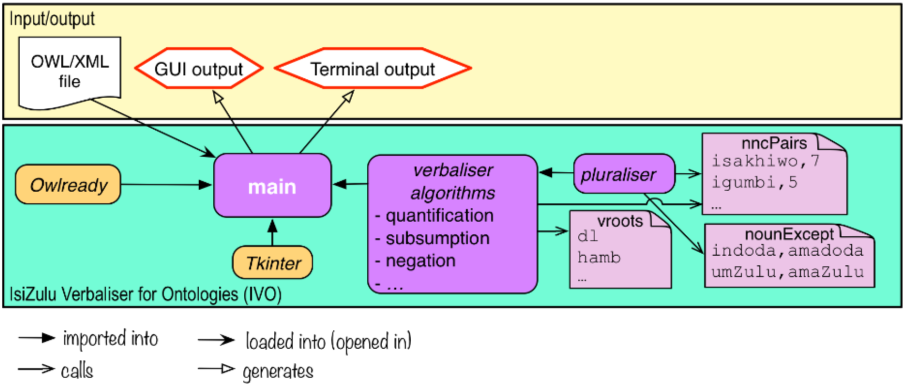
Evaluation

- Typical way of evaluating: ask linguists and/or intended target group
- Questions depend on what you want to know; e.g.,
    - Does the text capture the semantics adequately?
    - Must it really be grammatically correct or is understandable also acceptable?
    - Compared against alternate representation (figures, tables) or human-authored text?
- Survey, asked linguists and non-linguists for their preferences
- 10 questions pitting the patterns against each other
- Online, with isiZulu-localised version of Limesurvey

Evaluation – interesting results

- Linguist agreed more among each other than the 'non-linguists'
- More agreement for the shorter sentences
- Open questions on 'deep Zulu' vs 'township Zulu', level of education in isiZulu, dialects
  - Sociolinguistics is not our task to investigate, but it may affect human evaluation results w.r.t. quality, grammaticality, naturalness

Implementation (1/3)



Implementation (2/3)

```
#subsumption
def isa_zu(sub,super):
    if super.startswith('i'):
        return sub + ' y' + super
    elif super.startswith('a') or super.startswith('o') or super.startswith('u'):
        return sub + ' ng' + super
    else:
        return print('this is not a well-formed isiZulu noun.')
```

https://github.com/mkeet/GENIproject/

Implementation (2/3)

```python
#simple existential quantification
# modified cf zulurules to handle also vowel-commencing vroots
def exists_zu(sub,op,super):
    nc1m = find_nc(sub)
    nc2m = find_nc(super)
    pl = plural_zu(sub,nc1m)
    nc2 = strip_m(nc2m)
    ncp = look_ncp(nc1m)
    qca = look_qca(ncp)
    rc = look_relc(nc2)
    qc = look_qce(nc2)
    rt = find_rt(op)
    if rt[0] in 'aeiou':
        conjugrt = sc_vowel_vroot(rt,ncp)
    else:
        sc = look_sc(ncp)
        conjugrt = sc + rt
    return qca + ' ' + pl + ' ' + conjugrt + 'a' + ' ' + super + ' ' + rc + qc + 'dwa'
```

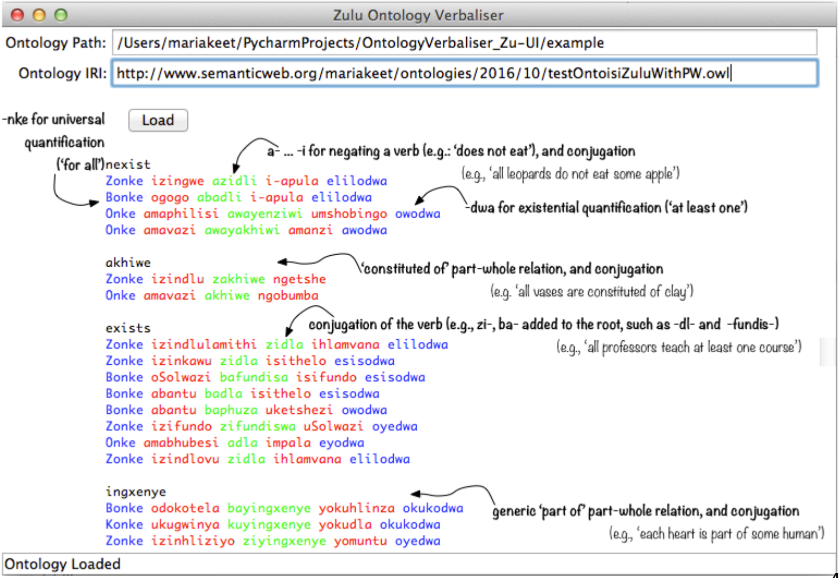
https://github.com/mkeet/GENIproject/

Implementation (2/3)

```
<SubClassOf>
    <Class IRI="#indlovu"/>
    <Class IRI="#isilwane"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#indlovu"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#dla"/>
        <Class IRI="#ihlamvana"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
```

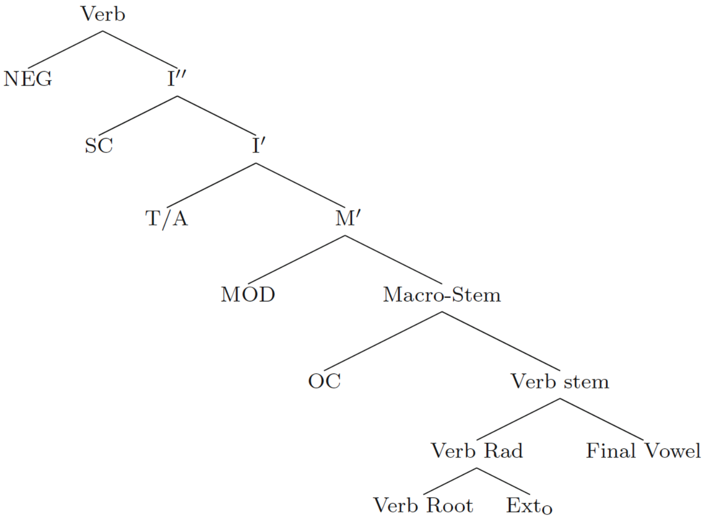
https://github.com/mkeet/GENIproject/

Sentences outputted as pretty printing or plaintext (3/3)



Outline

Figuring out the present tense



Figuring out the present tense [Keet and Khumalo(2017b)]

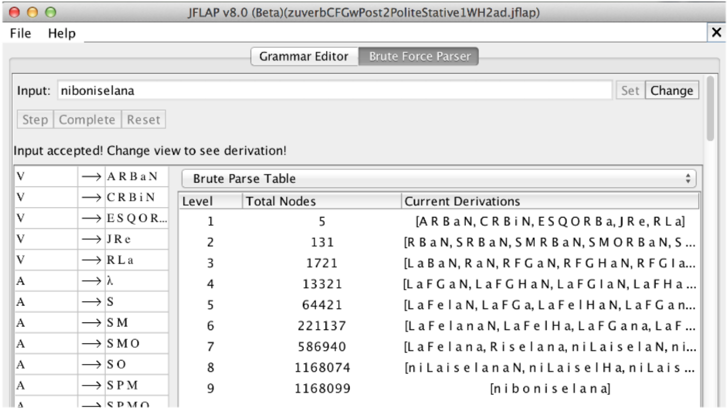
1. Verb, and start of the grammar:
   $v \rightarrow$ *pre vr post* a *wh* | *npre vr post* i *wh* | *ppre vr* e | *vr st* a | *excl s cont o vr post* a
2. Prefix (subject and object concord, tense, mode, and aspect):
   pre $\rightarrow$ *s* | *s m* | *s t m* | *s asp m* | *s o* | *s m o* | *s t m o* | *s asp m o*
3. Negative prefix (negation; e.g. 'does not' eat):
   npre $\rightarrow$ *ns* | *ns m* | *ns t m* | *ns asp m* | *ns o* | *ns m o* | *ns t m o* | *ns asp m o*
4. Postfix, begin the "CARP" extensions:
   post $\rightarrow$ *c* | *c a* | *c a r* | *c a p* | *c r* | *c r p* | *c p* | *c a r p* | *a* | *a r* | *a r p* | *a p* | *r* | *r p* | *p* | ε
5. List of subject concords and negative subject concords (terminals for conjugation):
   s $\rightarrow$ ngi | u | si | ni | ba | i | li | a | zi | lu | bu | ku | ε
   ns $\rightarrow$ angi | awu | aka | ali | asi | ayi | alu | abu | aku | ani | aba | awa | azi | ε
6. List of mod:
   m $\rightarrow$ a | e | ka | ma | nga | ε
7. List of tense (present (ε) and continuous (ya)tense; incomplete):
   t $\rightarrow$ ya | ε
8. List of aspect (additional rules omitted in this first iteration):
   asp $\rightarrow$ sa | se | be | ile | ε
9. List of object concords:
   o $\rightarrow$ ngi | si | ku | ni | m | ba | wu | yi | li | wa | zi | lu | bu | ε
10. Causative:
    c $\rightarrow$ is
11. Applicative:
    a $\rightarrow$ el
12. Reciprocative:
    r $\rightarrow$ an
13. Passive (with phonological conditioning options):
    p $\rightarrow$ iw | w
14. Politeness (own prefix system and a FV=e):
    ppre $\rightarrow$ *pl s*
    pl $\rightarrow$ aw | awu | mawu | ε | ma
15. Stative (insertion of the *-ek-* between the VR and the FV):
    st $\rightarrow$ ek
16. Wh-questions (in the post-final slot and are added at the end of the verb, being *-ni* 'what'/'who'/ 'why'/'how', *-nini* 'when', and *-phi* 'where'.):
    wh $\rightarrow$ ni | nini | phi | ε
17. 'Double aspect'/exclusive (with *excl* $\subset$ *asp*)
    excl $\rightarrow$ se
18. Continuous tense (with *cont* $\subset$ *t*):
    cont $\rightarrow$ ya
19. Lexicon of verb roots:
    vr $\rightarrow$ ab | ... | zwib

Evaluation – is the grammar any good?

- Does it generate a verb that is correct?
  (if yes: good)
- Does it generate incorrect verbs?
  (if yes: rules missing or a wrong rule)
- Can it not generate a correct verb?
  (if yes: rules missing or a wrong rule)
- Can it not generate an incorrect verb?
  (if yes: good)
- Does it recognise a correct verb?
  (if yes: good)
- Does it recognise incorrect verbs?
  (if yes: rules missing)
- Does it reject correct verbs?
  (if yes: rules missing or too restrictive)
- Does it reject incorrect verbs?
  (if yes: good)

Evaluation – this can be automated too

JFLAP `http://www.cs.duke.edu/csed/jflap/`



Evaluation – this can be automated too

JFLAP `http://www.cs.duke.edu/csed/jflap/`

Derivation Tree | Derivation Table

Production	Derivation
	V
V->A R B a N	A R B a N
A->S	S R B a N
S->n i	n i R B a N
R->b o n	n i b o n B a N
B->F G H	n i b o n F G H a N
F->i s	n i b o n i s G H a N
G->e l	n i b o n i s e l H a N
H->a n	n i b o n i s e l a n a N
N->λ	n i b o n i s e l a n a

Extensions: part-whole relations

- Part-whole relations are used widely in medical and healthcare ontologies
- Many different types (23 in OpenGalen)
- Would that be convenient 1:1 translations?

Extensions: part-whole relations

- Part-whole relations are used widely in medical and healthcare ontologies
- Many different types (23 in OpenGalen)
- Would that be convenient 1:1 translations?
  - No. both less and more specific ones: ontological differences
  - Other complications with verbs and prepositions
  - Details in: [Keet and Khumalo(2016)] [Keet(2017)] [Keet and Khumalo(2018)]

Further extensions and updates

- Adding (more) data-to-text to the knowledge-to-text
- Numbers [Mahlaza et al.(2022)], attributes ($\sim$ adjectives), etc. etc.
- Option: application-driven prioritization for what to look into
- Rules-based approach is a slow process

Further extensions and updates

- Adding (more) data-to-text to the knowledge-to-text
- Numbers [Mahlaza et al.(2022)], attributes ($\sim$ adjectives), etc. etc.
- Option: application-driven prioritization for what to look into
- Rules-based approach is a slow process
- A better architecture and system for the grammar, alike a 'SimpleNLG but then for NCB languages' (in the pipeline [Mahlaza and Keet(2021)])
- A better way to store the lexicon

What about ML and such for NLG?

- Feasibility of using machine learning or deep learning for templates:
  - Lack of good and relevant data (the bible and Ubuntu software manual are out-of-domain for healthcare messages)
  - Need comparatively more data (recall agglutination and type-to-token ratio)
  - Needs good NLU algorithms (POS tagging, morphological analysers)
  - Computing the language models is computationally expensive
  - The systems "hallucinates" and has spurious repetitions, in English at least
- Jan Buys commenced with that approach

Outline

Outline

Initial results

- Tried that in detail with Runyankore [Byamugisha et al.(2016)]: it's faster than starting from scratch; (also shown by [Bosch et al.(2008)] for morphological analysers)
- Multilingual pluraliser, with a new table for the noun classes to make it deterministic choices for computation [Byamugisha et al.(2018)]
- Trying to understand morphological and verb similarities as proxies for possibly [easy/not-easy] to bootstrap from/to [Keet(2016), Mahlaza and Keet(2018)]
- Assessing bootstrappability between vs across Guthrie zones w.r.t. ontology verbalisation; zones indeed are not a good predictor [Byamugisha(2019)]

Some practical ‘loose ends’

- Where to best store the NC info needed for ontology verbalisation?
- What if your language doesn’t have an ISO language tag?
- (There are more engineering questions to make it work)

Some practical 'loose ends'

- Where to best store the NC info needed for ontology verbalisation?
  - Ontolex-Lemon is good for declarative information, not for rules
  - Annotation model [Keet and Chirema(2016)]
  - And this for more NCB languages: WikiWorkshop 2022 abstract with a list of requirements: https://wikiworkshop.org/2022/papers/WikiWorkshop2022_paper_31.pdf
- What if your language doesn't have an ISO language tag?
- (There are more engineering questions to make it work)

Some practical 'loose ends'

- Where to best store the NC info needed for ontology verbalisation?
  - Ontolex-Lemon is good for declarative information, not for rules
  - Annotation model [Keet and Chirema(2016)]
  - And this for more NCB languages: WikiWorkshop 2022 abstract with a list of requirements: `https://wikiworkshop.org/2022/papers/WikiWorkshop2022_paper_31.pdf`
- What if your language doesn't have an ISO language tag?
  - Create your own!
  - e.g., with MoLA [Gillis-Webber et al.(2019)]
- (There are more engineering questions to make it work)

Outline

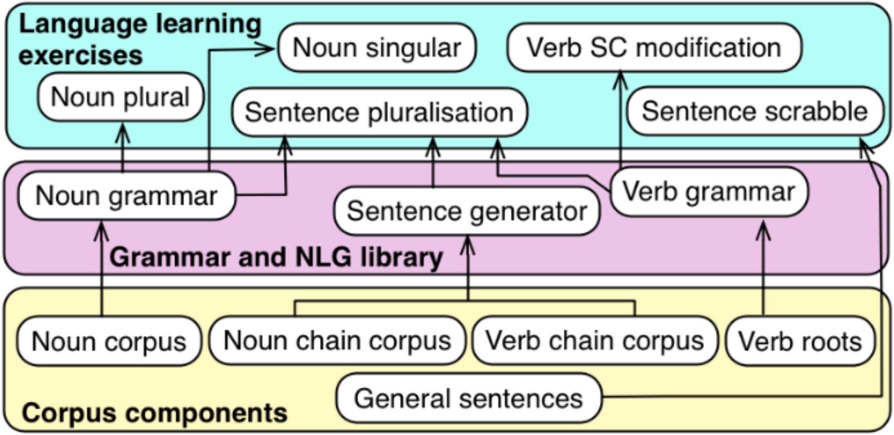
The NLG algorithms can be used elsewhere: CALL

- Paper-based language learning exercises
- Exercise books have a lot of exercises on ‘give plural noun’, ‘complete verb’ etc

The NLG algorithms can be used elsewhere: CALL

- Paper-based language learning exercises
- Exercise books have a lot of exercises on ‘give plural noun’, ‘complete verb’ etc
- Our CNL & NLG algorithms already can do that!
- Reuse the algorithms to pluralise and conjugate
- Proof of concept tool [Gilbert and Keet(2018)]; can generate 39501 unique question sentences of two or three words and compute their answers

Tool: architecture



Corpus? Yes, sort of

- Gather common words from text
- Create sets of relevant combinations of nouns and verbs
  - Transitive verbs require an object, intransitive ones do not
  - Refined with semantics; e.g., animals can eat, but furniture can't

Corpus? Yes, sort of

- Gather common words from text
- Create sets of relevant combinations of nouns and verbs
  - Transitive verbs require an object, intransitive ones do not
  - Refined with semantics; e.g., animals can eat, but furniture can't
- Examples:

Noun chain list
ubaba <1a> washa;sula;faka;khuluma
umzali <1;s> ALL_v;e_dumisa;e_cisha

Verb chain list
washa <t> imoto;umshini;umnyango
sula <> ifasitela;imoto;ipuleti
khuluma <t> ALL_1;ALL_1a

nouns may take some verbs, all verbs (ALL_v) with or without exceptions (e_), verbs with some specified classes (e.g., ALL_1) or with nouns in specific noun classes only, such as all people (NC1, NC1a)

Examples of the CNL it uses

- Pluralise subject

  Q: ** Umfowethu bayaphuza*

  A: *Abafowethu bayaphuza*

  [prefixSG+stem] [PLSC+VerbRoot+FV]

  [prefixPL+stem] [PLSC+VerbRoot+FV]

Examples of the CNL it uses

- Pluralise subject

  Q: ** Umfowethu bayaphuza*

  A: *Abafowethu bayaphuza*

  [prefixSG+stem] [PLSC+VerbRoot+FV]

  [prefixPL+stem] [PLSC+VerbRoot+FV]

- Negate the verb

  Q: *Batotoba*

  A: *Abatotobi*

  [PLSC+VerbRoot+FV]

  [PLNEGSC+VerbRoot+NEGFV]

Examples of the CNL it uses

- Pluralise subject

  Q: ** Umfowethu bayaphuza*

  A: *Abafowethu bayaphuza*

  [prefixSG+stem] [PLSC+VerbRoot+FV]

  [prefixPL+stem] [PLSC+VerbRoot+FV]

- Negate the verb

  Q: *Batotoba*

  A: *Abatotobi*

  [PLSC+VerbRoot+FV]

  [PLNEGSC+VerbRoot+NEGFV]

- Possible to combine components for new exercises

  [prefixSG+stem] [SGSC+VerbRoot+FV] [prefixSG+stem]

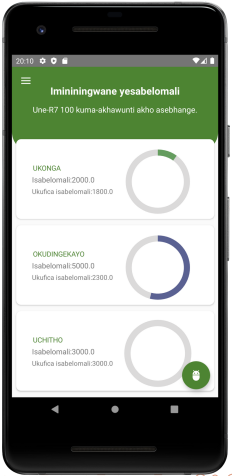
  [prefixPL+stem] [PLNEGSC+VerbRoot+NEGFV] [prefixPL+stem]

  Q: *umfowethu usula inkomishi* '(my) brother washes the cup'

  A: *abafowethu abasuli izinkomishi* '(my) brothers do not wash the cups'

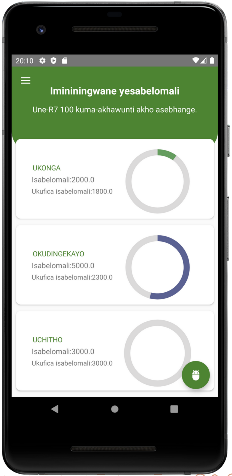
From indo-arabic numerals to text

- Financial illiteracy wrt personal finances, exclusion wrt languages banks offer their services in, lack of digital resources
- Design an app for that[1]
- Link that to TTS



From indo-arabic numerals to text

- Financial illiteracy wrt personal finances, exclusion wrt languages banks offer their services in, lack of digital resources
- Design an app for that[1]
- Link that to TTS
- Those pills in patient discharge notes; *amaphilisi ayishumi* 'ten pills'
- New feedback in CALL systems; *Uqede imisebenzi eziyishumi* 'You completed ten exercises'
- First results in [Mahlaza et al.(2022)]



Outline

Summary

- Computational view on NCB languages, on CNL and NLG
- Resulted in novelties *both* in computing *and* in linguistics
- Toward a tailor-made grammar engine for surface realisation, with customisable templates
- NLG algorithms generic and modularised in the sense that they can be reused in other tools (CALL exercises)
- Low resource languages a challenge for both rule-based and data-driven approaches, but in different ways; take your pick

Collaborators and Funding

- IsiZulu Linguist: Langa Khumalo
- Current/former students: Joan Byamugisha, Catherine Chavula, Takunda Chirema, Nikhil Gilbert, Francis Gillis-Webber, Zola Mahlaza, Sindiso Mkhatshwa, Junior Moraba, Gerald Ngumbulu, Musa Xakaza

- NRF CPRR grants: GeNI & MoRENL
  http://www.meteck.org/files/geni/
  http://www.meteck.org/MoReNL/

References I

Sonja Bosch, Laurette Pretorius, and Axel Fleisch.
Experimental bootstrapping of morphological analysers for nguni languages.
Nordic Journal of African Studies, 17(2):66–88, 2008.

J. Byamugisha, C.M. Keet, and B. DeRenzi.
Bootstrapping a runyankore cnl from an isizulu cnl.
In B. Davis et al., editors, *5th Workshop on Controlled Natural Language (CNL'16)*, volume 9767 of *LNAI*, pages 25–36. Springer, 2016.
25-27 July 2016, Aberdeen, UK.

J. Byamugisha, C. M. Keet, and B. DeRenzi.
Pluralizing nouns across agglutinating Bantu languages.
In *27th International Conference on Computational Linguistics (COLING'18)*, pages 2633–2643. ACL, 2018.
20-26 August, 2018, Santa Fe, New Mexico, USA.

Joan Byamugisha.
Ontology Verbalization in Agglutinating Bantu Languages: A Study of Runyankore and Its Generalizability.
Phd thesis, Department of Computer Science, November 2019 2019.

A. Gatt and E. Reiter.
Simplenlg: A realisation engine for practical applications.
In E. Krahmer and M. Theune, editors, *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG'09)*, pages 90–93. ACL, 2009.
March 30-31, 2009, Athens, Greece.

References II

Nikhil Gilbert and C. Maria Keet.
Automating question generation and marking of language learning exercises for isiZulu.
In Brian Davis, C. Maria Keet, and Adam Wyner, editors, *6th International Workshop on Controlled Natural language (CNL'18)*, volume 304 of *FAIA*, pages 31–40. IOS Press, 2018.
Co. Kildare, Ireland, 27-28 August 2018.

F. Gillis-Webber, S. Tittel, and C. M. Keet.
A model for language annotations on the web.
In *1st Iberoamerican conference on Knowledge Graphs and Semantic Web (KGSWC'19)*, volume 1029 of *CCIS*, pages 1–16. Springer, 2019.
24-28 June 2019, Villa Clara, Cuba.

Mustafa Jarrar, C. Maria Keet, and Paolo Dongilli.
Multilingual verbalization of ORM conceptual models and axiomatized ontologies.
Starlab technical report, Vrije Universiteit Brussel, Belgium, February 2006.
URL `http://www.meteck.org/files/ORMmultiverb_JKD.pdf`.

C. M. Keet.
An assessment of orthographic similarity measures for several african languages.
Technical Report Arxiv.org 1608.03065, University of Cape Town, August 2016.
URL `http://arxiv.org/abs/1608.03065`.

C. M. Keet.
Representing and aligning similar relations: parts and wholes in isizulu vs english.
In J. Gracia, F. Bond, J. McCrae, P. Buitelaar, C. Chiarcos, and S. Hellmann, editors, *Language, Data, and Knowledge 2017 (LDK'17)*, volume 10318 of *LNAI*, pages 58–73. Springer, 2017.
19-20 June, 2017, Galway, Ireland.

References III

C. M. Keet and T. Chirema.
A model for verbalising relations with roles in multiple languages.
In E. Blomqvist, P. Ciancarini, F. Poggi, and F. Vitali, editors, *Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'16)*, volume 10024 of *LNAI*, pages 384–399. Springer, 2016.
19-23 November 2016, Bologna, Italy.

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

C. M. Keet and L. Khumalo.
Grammar rules for the isizulu complex verb.
Southern African Journal of Language and Linguistics, 35(2):183–200, 2017b.

C. M. Keet and L. Khumalo.
On the ontology of part-whole relations in Zulu language and culture.
In S. Borgo and P. Hitzler, editors, *10th International Conference on Formal Ontology in Information Systems 2018 (FOIS'18)*, volume 306 of *FAIA*, pages 225–238. IOS Press, 2018.
17-21 September, 2018, Cape Town, South Africa.

C. Maria Keet and Langa Khumalo.
Toward verbalizing logical theories in isiZulu.
In B. Davis, T. Kuhn, and K. Kaljurand, editors, *Proceedings of the 4th Workshop on Controlled Natural Language (CNL'14)*, volume 8625 of *LNAI*, pages 78–89. Springer, 2014a.
20-22 August 2014, Galway, Ireland.

References IV

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, 2014b.
August 18-20, 2014, Prague, Czech Republic.

C. Maria Keet and Langa Khumalo.
On the verbalization patterns of part-whole relations in isizulu.
In *9th International Natural Language Generation conference (INLG'16)*, pages 174–183. ACL, 2016.
5-8 September, 2016, Edinburgh, UK.

Tobias Kuhn.
A principled approach to grammars for controlled natural languages and predictive editors.
Journal of Logic, Language and Information, 22(1):33–70, 2013.

Z. Mahlaza and C. Maria Keet.
Formalisation and classification of grammar and template-mediated techniques to model and ontology verbalisation.
Int. J. Metadata, Semantics and Ontologies, 14(3):249–262, 2022.

Z. Mahlaza and C.M. Keet.
Toct: A task ontology to manage complex templates.
In Emilio M. Sanfilippo et al., editors, *FOIS 2021 Ontology Showcase, The Joint Ontology Workshops (JOWO'21)*, volume 2969 of *CEUR-WS*, 2021.

References V

Zola Mahlaza and C. Maria Keet.
Measuring verb similarity using binary coefficients with application to isixhosa and isizulu.
In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT'18)*, pages 65–71, New York, NY, USA, 2018. ACM.
ISBN 978-1-4503-6647-2.
doi: 10.1145/3278681.3278690.
URL http://doi.acm.org/10.1145/3278681.3278690.

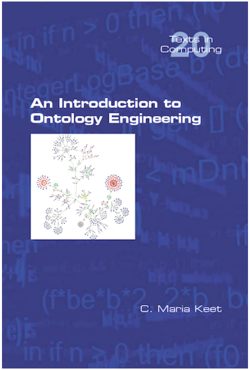
Zola Mahlaza, Junior Moraba, C. Maria Keet, and L. Khumalo.
Automatically generating isizulu words from indo-arabic numeralst.
In *Proceedings of the South African Institute of Computer Scientists and Information Technologists (SAICSIT'22)*, SAICSIT. EPIC, 2022.
URL http://doi.acm.org/10.1145/3129416.3129443.

K. Winkler, T. Kuhn, and M Volk.
Evaluating the fully automatic multi-language translation of the swiss avalanche bulletin.
In B. Davis, T. Kuhn, and K. Kaljurand, editors, *Proceedings of the 4th Workshop on Controlled Natural Language (CNL'14)*, volume 8625 of *LNAI*, pages 44–54. Springer, 2014.
20-22 August 2014, Galway, Ireland.

Thank you!

Questions?

My award-winning textbook



A memoir