Knowledge-to-text Natural Language Generation for Agglutinating African Languages

C. Maria Keet

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

Tech Talk at Google, Zurich, Switzerland
Wikimedia Foundation Google.org Fellows offsite workshop, 23-26 August 2022
Knowledge engineering team: http://www.meteck.org/keen/

Aim: to contribute computing theory, methods, and techniques to the knowledge society

Scope is knowledge engineering in its broad sense, including ontology engineering, the Semantic Web, intelligent (logic-based, ontology-driven) conceptual modelling, and natural language generation
Outline

1 Motivation
   - Context
   - Notes on NCB languages

2 Rule-based NLG
   - CNL and NLG in a nutshell
   - Generating basic sentences in isiZulu
   - Extending basic sentences

3 Discussion

4 Summary
1 Motivation
   • Context
   • Notes on NCB languages

2 Rule-based NLG
   • CNL and NLG in a nutshell
   • Generating basic sentences in isiZulu
   • Extending basic sentences

3 Discussion

4 Summary
Outline

1. Motivation
   - Context
     - Notes on NCB languages

2. Rule-based NLG
   - CNL and NLG in a nutshell
   - Generating basic sentences in isiZulu
   - Extending basic sentences

3. Discussion

4. Summary
Motivation

- >1.4 billion people in Africa, most do or can speak a language other than English or French
  - South Africa: isiZulu and isiXhosa most widely spoken languages, by first language speakers
  - 23% or about 11 million people isiZulu, 8 million (isiXhosa)
Motivation

>1.4 billion people in Africa, most do or can speak a language other than English or French
- 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”)

... but there is very limited ICT in/for African languages of the Niger-Congo family, and only for a few languages
Motivation

- NLP tools also for African languages proper
- Requires tools with African languages in at least the interface, not just some ‘pretty pictures and icons’
- A.o.t., 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:**

```
  <owl:Restriction>
  </owl:Restriction>
</owl:Class>
```

**JSON:**

```json
{  "name": "Dependency",  "participants": [    {      "name": "Employee",      "role": "provides_for",      "participation": "strong",      "min": "0",      "max": "N"    },    {      "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>
  </PLANT>
</CATALOG>
```
Structured sentences – examples for knowledge-to-text

- Electronic health records and patient discharge notes generation

- Requirements engineering and CQs for app development

- 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”

- Requirements engineering and CQs for app development
  - Capture and validate relevant business logic
    - “Who works for the HR Department?”

- 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”

- Requirements engineering and CQs for app development
  - Capture and validate relevant business logic
    - “Who works for the HR Department?”

- 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

- Rule-based Controlled Natural Languages & Natural Language Generation
- Knowledge-to-text; input: ontologies, knowledge graphs etc
- Agglutinating Niger-Congo B languages (aka ‘bantu languages’)
Outline

1 Motivation
   • Context
   • Notes on NCB languages

2 Rule-based NLG
   • CNL and NLG in a nutshell
   • Generating basic sentences in isiZulu
   • Extending basic sentences

3 Discussion

4 Summary
Basics

1500-2000 African languages (6 main groups) spoken by 1.4 billion people

Core characteristics relevant for computation (1/2)

1. 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
<table>
<thead>
<tr>
<th>NC</th>
<th>AU</th>
<th>PRE</th>
<th>Stem (example)</th>
<th>Meaning</th>
<th>Example (isiZulu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u-</td>
<td>m(u)-</td>
<td>fana</td>
<td>humans and other animates</td>
<td>umfana</td>
</tr>
<tr>
<td>2</td>
<td>a-</td>
<td>ba-</td>
<td>fana</td>
<td>boy</td>
<td>abafana</td>
</tr>
<tr>
<td>1a</td>
<td>u-</td>
<td>-</td>
<td>baba</td>
<td>kinship terms and proper names</td>
<td>ubaba</td>
</tr>
<tr>
<td>2a</td>
<td>o-</td>
<td>-</td>
<td>baba</td>
<td>fathers</td>
<td>obaba</td>
</tr>
<tr>
<td>3a</td>
<td>u-</td>
<td>-</td>
<td>-shizi</td>
<td>nonhuman</td>
<td>ushizi</td>
</tr>
<tr>
<td>2a</td>
<td>o-</td>
<td>-</td>
<td>-shizi</td>
<td>cheese</td>
<td>oshizi</td>
</tr>
<tr>
<td>3</td>
<td>u-</td>
<td>mi-</td>
<td>m(u)-fula</td>
<td>trees, plants, non-paired body parts</td>
<td>umfula</td>
</tr>
<tr>
<td>4</td>
<td>i-</td>
<td>mi-</td>
<td>fula</td>
<td>river</td>
<td>imifula</td>
</tr>
<tr>
<td>5</td>
<td>i-</td>
<td>(li)-</td>
<td>-gama</td>
<td>fruits, paired body parts, and natural phenomena</td>
<td>igama</td>
</tr>
<tr>
<td>6</td>
<td>a-</td>
<td>ma-</td>
<td>gama</td>
<td>name</td>
<td>amagama</td>
</tr>
<tr>
<td>7</td>
<td>i-</td>
<td>si-</td>
<td>-hlalo</td>
<td>inanimates and manner/style</td>
<td>isihlalo</td>
</tr>
<tr>
<td>8</td>
<td>i-</td>
<td>zi-</td>
<td>-hlalo</td>
<td>chair</td>
<td>izihlalo</td>
</tr>
<tr>
<td>9a</td>
<td>i-</td>
<td>ma-</td>
<td>-rabha</td>
<td>nonhuman</td>
<td>irabha</td>
</tr>
<tr>
<td>6</td>
<td>a-</td>
<td>-</td>
<td>-rabha</td>
<td>rubber</td>
<td>amarabha</td>
</tr>
<tr>
<td>9</td>
<td>i(n)-</td>
<td>-</td>
<td>-ja</td>
<td>animals</td>
<td>inja</td>
</tr>
<tr>
<td>10</td>
<td>i-</td>
<td>zi(n)-</td>
<td>-ja</td>
<td>dog</td>
<td>izinja</td>
</tr>
<tr>
<td>11</td>
<td>u-</td>
<td>(lu)-</td>
<td>-thi</td>
<td>inanimates and long thin objects</td>
<td>uthi</td>
</tr>
<tr>
<td>10</td>
<td>i-</td>
<td>zi(n)-</td>
<td>-thi</td>
<td>stick</td>
<td>izinthi</td>
</tr>
<tr>
<td>14</td>
<td>u-</td>
<td>bu-</td>
<td>-hle</td>
<td>abstract nouns</td>
<td>ubuhle</td>
</tr>
<tr>
<td>15</td>
<td>u-</td>
<td>ku-</td>
<td>-cula</td>
<td>infinitives</td>
<td>ukucula</td>
</tr>
<tr>
<td>17</td>
<td>ku-</td>
<td></td>
<td>locatives, remote/ general</td>
<td>locative</td>
<td></td>
</tr>
</tbody>
</table>
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 OC)-(NC1 OC)-VR-App-FV-Loc-Emp-Dec
Illustrative examples of some consequences (isiZulu)

- ‘and’, enumerative: *na-*, phonologically conditioned
  Ex: milk and butter: *ubisi n'ebhotela*  
  (-a+i=-e-)
  Ex: butter and milk: *ibhotela n'obisi*  
  (-a+u=-o-)

Verbs: concordial agreement (∼ conjugation) based on noun class
Ex: The human eats 
Ex: The dog eats

‘is not a’: combine NEG SC with PRON, both depend on nc
Ex: an animal is not a plant: *isilwane asiwona* umuthi  
Ex: a plant is not an animal: *umuthi awusona* isilwane
Illustrative examples of some consequences (isiZulu)

- ‘and’, enumerative: *na-*-, phonologically conditioned
  Ex: milk and butter: *ubisi nebhotela* (-a+i=−e−)
  Ex: butter and milk: *ibhotela nobisi* (-a+u=−o−)

- Verbs: concordial agreement (∼ conjugation) based on noun class
  Ex: The human eats *umuntu udla*
  Ex: The dog eats *inja idla*
Illustartive examples of some consequences (isiZulu)

- ‘and’, enumerative: $na$-, phonologically conditioned
  - Ex: milk and butter: $ubisi$ $nебhotela$ (-a+i=-e-)
  - Ex: butter and milk: $ibhotela$ $nobisi$ (-a+u=-o-)

- Verbs: concordial agreement ($\sim$ conjugation) based on noun class
  - Ex: The human eats $umuntu$ $udla$
  - Ex: The dog eats $inja$ $idla$

- ‘is not a’: combine NEG SC with PRON, both depend on nc
  - Ex: an animal is not a plant: $isilwane$ $asiwona$ $umuthi$
  - Ex: a plant is not an animal: $umuthi$ $awusona$ $isilwane$
Concordial agreement

3. System of concordial agreement


‘The little boys will buy the big books’
Outline

1. Motivation
   - Context
   - Notes on NCB languages

2. Rule-based NLG
   - CNL and NLG in a nutshell
   - Generating basic sentences in isiZulu
   -Extending basic sentences

3. Discussion

4. Summary
Outline

1 Motivation
   - Context
   - Notes on NCB languages

2 Rule-based NLG
   - CNL and NLG in a nutshell
   - Generating basic sentences in isiZulu
   - Extending basic sentences

3 Discussion

4 Summary
Short answer

- **C**ontrolled **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**

**Chest Pain**

Have you had any recent pain in your chest? - Uke waba nobuhlungu esifubeni maduzane?

Does the pain radiate to your jaw, neck or arm? - Engabe ubuhlungu bakho bujikeleza emihlathini, emqaleni noma nasezingalweni?

Does anything precipitate or relieve the pain? - Ingabe ikhona into eyenza ubuhlungu buqhubeka noma eyehlisa ubuhlungu?

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

![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.](image-url)
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

Each [C1] [R1] at least one [C2]
With logic-based reconstruction

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

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

DL: **Course** $\sqsubseteq \exists \text{is_teaches}.\text{Professor}

- mandatory constraint / existential quantification (all-some pattern)
- **Each** [C1] [R1] **at least one** [C2]
ORM model snippet, serialised in XML

...<Predicate>
<Object_Role ID='ExEN:249' Object='Professor' Role='teaches'/>
<Object_Role ID='ExEN:250' Object='Course' Role='taught'/>
</Predicate>
...

<Constraint xsi:type='Mandatory'>
<Object_Role>ExEN:249</Object_Role>
</Constraint>
...

Example of static templates in ES and EN

Simple existential quantification (‘mandatory constraint’) template

Each \([C1] \ [R1] \text{ at least one } [C2]\)

for a large fragment of ORM, and 11 languages [Jarrar et al.(2006)]
Example of static templates in ES and EN

Simple existential quantification (‘mandatory constraint’) template

**Each [C1] [R1] at least one [C2]**

```xml
<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 in ES and EN

Simple existential quantification (‘mandatory constraint’) template
Each [C1] [R1] at least one [C2]

for a large fragment of ORM, and 11 languages [Jarrar et al.(2006)]
Simple existential quantification (‘mandatory constraint’) template

Each [C1] [R1] at least one [C2]

for a large fragment of ORM, and 11 languages [Jarrar et al.(2006)]
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
- Same stems or words and core structure of the grammar-infused template, generate different sentences based on grammatical features declared
  - yes/no pronomial, present/past tense, gender
Somewhat 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)

\[
\begin{align*}
\text{Sentence} & \rightarrow \text{NounPhrase} \mid \text{VerbPhrase} \\
\text{NounPhrase} & \rightarrow \text{Adjective} \mid \text{NounPhrase} \\
\text{NounPhrase} & \rightarrow \text{Noun} \\
\text{Noun} & \rightarrow \text{car} \mid \text{train} \\
\text{Adjective} & \rightarrow \text{big} \mid \text{broken} \\
\end{align*}
\]

+ 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

```xml
<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
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
  - e.g., such as [Kuhn(2013)], Grammatical Framework (http://www.grammaticalframework.org/), SimpleNLG [Gatt and Reiter(2009)]
- Different ways to mix ‘simple’ static templates with grammar rules [Mahlaza and Keet(2020)]
The ‘NLG pipeline’

1. Content determination
2. Discourse planning
3. Sentence aggregation
4. Lexicalisation
5. Referring expression generation
6. Linguistic realisation

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)

(based on [Reiter and Dale(1997)])
Outline

1 Motivation
   - Context
   - Notes on NCB languages

2 Rule-based NLG
   - CNL and NLG in a nutshell
   - Generating basic sentences in isiZulu
   - Extending basic sentences

3 Discussion

4 Summary
Question

Can we use any of the simple template-based approaches for agglutinating Niger-Congo B languages?
Question

- Can we use any of the simple template-based approaches for agglutinating Niger-Congo B languages?
  - It depends... but mostly: no
Question

- Can we use any of the simple template-based 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 the simple template-based 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
Ontology verbalisation

The NLG ‘pipeline’

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.)

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)
**ALC syntax (a popular description logic)**

- **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., \( \text{Lion} \sqsubseteq \exists \text{eats.\text{Herbivore}} \sqcap \forall \text{eats.\text{Herbivore}} \)
\textbf{ALC} semantics

- \textit{domain of interpretation}, and an \textit{interpretation}, where:
  - Domain $\Delta$ is a non-empty set of objects
  - Interpretation: $\cdot^\mathcal{I}$ is the \textit{interpretation function}, domain $\Delta^\mathcal{I}$
    - $\cdot^\mathcal{I}$ maps every concept name $A$ to a subset $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$
    - $\cdot^\mathcal{I}$ maps every role name $R$ to a subset $R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$
    - $\cdot^\mathcal{I}$ maps every individual name $a$ to elements of $\Delta^\mathcal{I}$: $a^\mathcal{I} \in \Delta^\mathcal{I}$
  - Note: $\top^\mathcal{I} = \Delta^\mathcal{I}$ and $\bot^\mathcal{I} = \emptyset$
- $(-C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$
- $(C \cap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$
- $(C \cup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$
- $(\forall R.C)^\mathcal{I} = \{x \mid \forall y. R^\mathcal{I}(x, y) \rightarrow C^\mathcal{I}(y)\}$
- $(\exists R.C)^\mathcal{I} = \{x \mid \exists y. R^\mathcal{I}(x, y) \land C^\mathcal{I}(y)\}$
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 ⊑

(U1) Boy ⊑ ...
  wonke umfana ... (‘each boy...’; u- + -onke)
  bonke abafana ... (‘all boys...’; ba- + -onke)

(U2) Phone ⊑ ...
  lonke ifoni ... (‘each phone...’; li- + -onke)
  onke amafoni ... (‘all phones...’; a- + -onke)
<table>
<thead>
<tr>
<th>NC</th>
<th>QC (all)</th>
<th>QC&lt;sub&gt;oral+onke&lt;/sub&gt;</th>
<th>QC&lt;sub&gt;nke&lt;/sub&gt;</th>
<th>NEG SC</th>
<th>PRON</th>
<th>RC</th>
<th>QC&lt;sub&gt;dwa&lt;/sub&gt;</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u-onke → wonke</td>
<td>wo</td>
<td>aka-</td>
<td></td>
<td>yena</td>
<td>o-</td>
<td>ye-</td>
<td>mu-</td>
</tr>
<tr>
<td>2</td>
<td>ba-onke → bonke</td>
<td>bo</td>
<td>aba-</td>
<td></td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>1a</td>
<td>u-onke → wonke</td>
<td>wo</td>
<td>aka-</td>
<td></td>
<td>yena</td>
<td>o-</td>
<td>ye-</td>
<td>mu-</td>
</tr>
<tr>
<td>2a</td>
<td>ba-onke → bonke</td>
<td>bo</td>
<td>aba-</td>
<td></td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>3a</td>
<td>u-onke → wonke</td>
<td>wo</td>
<td>aka-</td>
<td></td>
<td>wona</td>
<td>o-</td>
<td>ye-</td>
<td>mu-</td>
</tr>
<tr>
<td>(2a)</td>
<td>ba-onke → bonke</td>
<td>bo</td>
<td>aba-</td>
<td></td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>3</td>
<td>u-onke → wonke</td>
<td>wo</td>
<td>ak-</td>
<td></td>
<td>wona</td>
<td>o-</td>
<td>wo-</td>
<td>mu-</td>
</tr>
<tr>
<td>4</td>
<td>i-onke → yonke</td>
<td>yo</td>
<td>ayi-</td>
<td></td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
<td>mi-</td>
</tr>
<tr>
<td>5</td>
<td>li-onke → lonke</td>
<td>lo</td>
<td>ali-</td>
<td></td>
<td>lona</td>
<td>eli-</td>
<td>lo-</td>
<td>li-</td>
</tr>
<tr>
<td>6</td>
<td>a-onke → onke</td>
<td>o</td>
<td>awa-</td>
<td></td>
<td>wona</td>
<td>a-</td>
<td>wo-</td>
<td>ma-</td>
</tr>
<tr>
<td>7</td>
<td>si-onke → sonke</td>
<td>so</td>
<td>asi-</td>
<td></td>
<td>sona</td>
<td>esi-</td>
<td>so-</td>
<td>si-</td>
</tr>
<tr>
<td>8</td>
<td>zi-onke → zonke</td>
<td>so</td>
<td>azi-</td>
<td></td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
<td>zi-</td>
</tr>
<tr>
<td>9a</td>
<td>i-onke → yonke</td>
<td>yo</td>
<td>ayi-</td>
<td></td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
<td>yi-</td>
</tr>
<tr>
<td>(6)</td>
<td>a-onke → onke</td>
<td>o</td>
<td>awa-</td>
<td></td>
<td>wona</td>
<td>a-</td>
<td>wo-</td>
<td>ma-</td>
</tr>
<tr>
<td>9</td>
<td>i-onke → yonke</td>
<td>yo</td>
<td>ayi-</td>
<td></td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
<td>yi-</td>
</tr>
<tr>
<td>10</td>
<td>z-i-onke → zonke</td>
<td>zo</td>
<td>azi-</td>
<td></td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
<td>zi-</td>
</tr>
<tr>
<td>11</td>
<td>lu-onke → lonke</td>
<td>lo</td>
<td>alu-</td>
<td></td>
<td>lona</td>
<td>olu-</td>
<td>lo-</td>
<td>lu-</td>
</tr>
<tr>
<td>(10)</td>
<td>zi-onke → zonke</td>
<td>zo</td>
<td>azi-</td>
<td></td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
<td>zi-</td>
</tr>
<tr>
<td>14</td>
<td>ba-onke → bonke</td>
<td>bo</td>
<td>abu-</td>
<td></td>
<td>bona</td>
<td>obu-</td>
<td>bo-</td>
<td>bu-</td>
</tr>
<tr>
<td>15</td>
<td>ku-onke → konke</td>
<td>zo</td>
<td>aku-</td>
<td></td>
<td>khona</td>
<td>oku-</td>
<td>zo-</td>
<td>ku-</td>
</tr>
<tr>
<td>NC</td>
<td>QCoral</td>
<td>QC (all)</td>
<td>QCnke</td>
<td>NEG SC</td>
<td>PRON</td>
<td>RC</td>
<td>QCdwa</td>
<td>EC</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>1</td>
<td>u-onke</td>
<td>wonke</td>
<td>wo-</td>
<td>aka-</td>
<td>yena</td>
<td>o-</td>
<td>ye-</td>
<td>mu-</td>
</tr>
<tr>
<td>2</td>
<td>ba-onke</td>
<td>bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>aba-</td>
<td>ba-</td>
<td></td>
</tr>
<tr>
<td>1a</td>
<td>u-onke</td>
<td>wonke</td>
<td>wo-</td>
<td>aka-</td>
<td>yena</td>
<td>o-</td>
<td>ye-</td>
<td>mu-</td>
</tr>
<tr>
<td>2a</td>
<td>ba-onke</td>
<td>bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>3a</td>
<td>u-onke</td>
<td>wonke</td>
<td>wo-</td>
<td>aka-</td>
<td>wona</td>
<td>o-</td>
<td>ye-</td>
<td>mu-</td>
</tr>
<tr>
<td>(2a)</td>
<td>ba-onke</td>
<td>bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>3</td>
<td>u-onke</td>
<td>wonke</td>
<td>wo-</td>
<td>awu-</td>
<td>wona</td>
<td>o-</td>
<td>wo-</td>
<td>mu-</td>
</tr>
<tr>
<td>4</td>
<td>i-onke</td>
<td>yonke</td>
<td>yo-</td>
<td>ayi-</td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
<td>mi-</td>
</tr>
<tr>
<td>5</td>
<td>li-onke</td>
<td>lonke</td>
<td>lo-</td>
<td>ali-</td>
<td>lona</td>
<td>eli-</td>
<td>lo-</td>
<td>li-</td>
</tr>
<tr>
<td>6</td>
<td>a-onke</td>
<td>onke</td>
<td>o-</td>
<td>awa-</td>
<td>wona</td>
<td>a-</td>
<td>wo-</td>
<td>ma-</td>
</tr>
<tr>
<td>7</td>
<td>si-onke</td>
<td>sonke</td>
<td>so-</td>
<td>asi-</td>
<td>sona</td>
<td>esi-</td>
<td>so-</td>
<td>si-</td>
</tr>
<tr>
<td>8</td>
<td>zi-onke</td>
<td>zonke</td>
<td>zo-</td>
<td>azi-</td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
<td>zi-</td>
</tr>
<tr>
<td>9a</td>
<td>i-onke</td>
<td>yonke</td>
<td>yo-</td>
<td>ayi-</td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
<td>yi-</td>
</tr>
<tr>
<td>(6)</td>
<td>a-onke</td>
<td>onke</td>
<td>o-</td>
<td>awa-</td>
<td>wona</td>
<td>a-</td>
<td>wo-</td>
<td>ma-</td>
</tr>
<tr>
<td>9</td>
<td>i-onke</td>
<td>yonke</td>
<td>yo-</td>
<td>ayi-</td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
<td>yi-</td>
</tr>
<tr>
<td>10</td>
<td>zi-onke</td>
<td>zonke</td>
<td>zo-</td>
<td>azi-</td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
<td>zi-</td>
</tr>
<tr>
<td>11</td>
<td>lu-onke</td>
<td>lonke</td>
<td>lo-</td>
<td>alu-</td>
<td>lona</td>
<td>olu-</td>
<td>lo-</td>
<td>lu-</td>
</tr>
<tr>
<td>(10)</td>
<td>zi-onke</td>
<td>zonke</td>
<td>zo-</td>
<td>azi-</td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
<td>zi-</td>
</tr>
<tr>
<td>14</td>
<td>ba-onke</td>
<td>bonke</td>
<td>bo-</td>
<td>abu-</td>
<td>bona</td>
<td>obu-</td>
<td>bo-</td>
<td>bu-</td>
</tr>
<tr>
<td>15</td>
<td>ku-onke</td>
<td>konke</td>
<td>zo-</td>
<td>aku-</td>
<td>khona</td>
<td>oku-</td>
<td>zo-</td>
<td>ku-</td>
</tr>
</tbody>
</table>
Subsumption (axiom pattern $A \sqsubseteq B$)

- 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 ngumuthi ('medicinal herb is a plant')
  - amakhambi yimithi ('medicinal herbs are plants')
  - wonke amakhambi ngumuthi ('all medicinal herbs are a plant')
(S2) (generic)
(S3) (determinate)
Possible subsumption patterns

a. $N_1 \langle$copulative $ng/y$ depending on first letter of $N_2\rangle N_2$.

b. $\langle$plural of $N_1\rangle \langle$copulative $ng/y$ depending on first letter of plural of $N_2\rangle \langle$plural of $N_2\rangle$.

c. $\langle$All-concord for NC$_x$\rangle onke $\langle$plural of $N_1$, being of NC$_x$\rangle$
   \langle$copulative $ng/y$ depending on first letter of $N_2\rangle N_2$. 
Existential Quantification (axiom pattern $A \sqsubseteq \exists R.B$)

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

yonke indlulamithi idla ihlamvana elilodwa

zonke izindlulamithi zidla ihlamvana elilodwa

('each giraffe eats at least one twig')

('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.
Walk-through of the algorithm

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

- \( \forall x \ (u\text{Solwazi}(x) \rightarrow \exists y \ (-\text{fundisa}(x, y) \land \text{Isifundo}(y))) \)
- \( u\text{Solwazi} \sqsubseteq \exists -\text{fundisa}.\text{Isifundo} \)
- ?
\[ \forall x \ (u\text{Solwazi}(x) \rightarrow \exists y \ (-\text{fundisa}(x, y) \land \text{Isifundo}(y))) \]

\[ u\text{Solwazi} \sqsubseteq \exists \ -\text{fundisa}\.\text{Isifundo} \]
∀x (uSolwazi(x) →

uSolwazi ∈ ∃-func

look-up NC

pluralise

for-all

Bonke oSolwazi

<table>
<thead>
<tr>
<th>NC</th>
<th>AU</th>
<th>PRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u-</td>
<td>m(u)-</td>
</tr>
<tr>
<td>2</td>
<td>a-</td>
<td>ba-</td>
</tr>
<tr>
<td>3a</td>
<td>u-</td>
<td>-</td>
</tr>
<tr>
<td>(2a)</td>
<td>o-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>i-</td>
<td>m(u)-</td>
</tr>
<tr>
<td>4</td>
<td>i-</td>
<td>mi-</td>
</tr>
<tr>
<td>5</td>
<td>i-</td>
<td>(li)-</td>
</tr>
<tr>
<td>6</td>
<td>a-</td>
<td>ma-</td>
</tr>
<tr>
<td>7</td>
<td>i-</td>
<td>si-</td>
</tr>
<tr>
<td>8</td>
<td>i-</td>
<td>zi-</td>
</tr>
<tr>
<td>9a</td>
<td>i-</td>
<td>-</td>
</tr>
<tr>
<td>(6)</td>
<td>a-</td>
<td>ma-</td>
</tr>
<tr>
<td>11</td>
<td>u-</td>
<td>(lu)-</td>
</tr>
<tr>
<td>(10)</td>
<td>i-</td>
<td>(zi(n)-</td>
</tr>
<tr>
<td>14</td>
<td>u-</td>
<td>bu-</td>
</tr>
<tr>
<td>15</td>
<td>u-</td>
<td>ku-</td>
</tr>
<tr>
<td>17</td>
<td>ku-</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NC</th>
<th>QC (all)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u-onke → wonke</td>
</tr>
<tr>
<td>2</td>
<td>ba-onke → bonke</td>
</tr>
<tr>
<td>3</td>
<td>u-onke → wonke</td>
</tr>
<tr>
<td>4</td>
<td>ba-onke → bonke</td>
</tr>
<tr>
<td>5</td>
<td>li-onke → lonke</td>
</tr>
<tr>
<td>6</td>
<td>a-onke → onke</td>
</tr>
<tr>
<td>7</td>
<td>si-onke → sonke</td>
</tr>
<tr>
<td>8</td>
<td>zi-onke → zonke</td>
</tr>
<tr>
<td>9a</td>
<td>i-onke → yonke</td>
</tr>
<tr>
<td>(6)</td>
<td>a-onke → onke</td>
</tr>
<tr>
<td>11</td>
<td>lu-onke → lonke</td>
</tr>
<tr>
<td>(10)</td>
<td>zi-onke → zonke</td>
</tr>
<tr>
<td>14</td>
<td>ba-onke → bonke</td>
</tr>
<tr>
<td>15</td>
<td>ku-onke → konke</td>
</tr>
</tbody>
</table>
\[ \forall x \ (u\text{Solwazi}(x) \rightarrow \exists y \ (\text{fundisa}(x, y, v) \land \text{Isifundo}(y))) \]

Bonke oSolwazi bafundisa

<table>
<thead>
<tr>
<th>NC</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u-ba-</td>
</tr>
<tr>
<td>2</td>
<td>u-ba-</td>
</tr>
<tr>
<td>1a</td>
<td>u-ba-</td>
</tr>
<tr>
<td>2a</td>
<td>u-ba-</td>
</tr>
<tr>
<td>3</td>
<td>u-ba-</td>
</tr>
<tr>
<td>4</td>
<td>u-i-</td>
</tr>
<tr>
<td>5</td>
<td>li-</td>
</tr>
<tr>
<td>6</td>
<td>a-</td>
</tr>
<tr>
<td>7</td>
<td>si-</td>
</tr>
<tr>
<td>8</td>
<td>zi-</td>
</tr>
<tr>
<td>9a</td>
<td>i-</td>
</tr>
<tr>
<td>9</td>
<td>a-</td>
</tr>
<tr>
<td>10</td>
<td>i-</td>
</tr>
<tr>
<td>11</td>
<td>zi-</td>
</tr>
<tr>
<td>14</td>
<td>lu-</td>
</tr>
<tr>
<td>15</td>
<td>zi-</td>
</tr>
<tr>
<td>17</td>
<td>bu-</td>
</tr>
<tr>
<td></td>
<td>ku-</td>
</tr>
</tbody>
</table>

reuse pluralised NC of subject

look-up SC of that NC

Bonke oSolwazi bafundisa
\[ \forall x \ (\text{uSolwazi}(x) \rightarrow \exists y \ (-\text{fundisa}(x, y) \land \text{Isifundo}(y))) \]

\[ \text{uSolwazi} \subseteq \exists \ -\text{fundisa}\ . \text{Isifundo} \]

Bonke oSolwazi bafundisa Isifundo
∀ᵧ (uSolwazi(ᵧ) → ∃ᵢ)

Bonke oSolwazi bafundisa Isifundo esisodwa
Generating basic sentences in isiZulu

**Rule-based NLG**

**English cf. isiZulu for the “all-some” pattern**

**Axiom type** ‘all-some’ ontology pattern (mandatory constraint)
\[
\forall x (X(x) \rightarrow \exists y (R(x, y) \land Y(y)))
\]
\[
X \sqsubseteq \exists R.Y
\]

**English**
- All [noun x pl.] [verb 3rd pers. pl.] at least one [noun y]
  - All professors teach at least one course
  - All professors write at least one book
  - All carnivores eat at least one animal
  - All elephants eat at least one apple

**IsiZulu**
- [QCall\textsubscript{nc\_x, pl\_x}] [noun \textsubscript{x\_nc\_x, pl\_x}] [SC\textsubscript{nc\_x, pl\_x}-verb] [noun \textsubscript{y\_nc\_y}] RC\textsubscript{nc\_y}-QC\textsubscript{nc\_y}-dwa
  - Bonke oSolwazi bafundisa isifundo esisodwa
  - Bonke oSolwazi babhala incwadi eyodwa
  - Onke amakhanivo adla isilwane esisolwa
  - Zonke izindlovu zidla i-apula elilodwa
Evaluation

- Typical way of evaluating: ask linguists and/or intended target group
- 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
Proof-of-concept implementation (1/3)

IsiZulu Verbaliser for Ontologies

Owlready

Tkinter

main

verbaliser

algorithms

- quantification
- subsumption
- negation
- ...

pluraliser

nncPairs

isakhiwo, 7

igumbi, 5

...

vroots

dl

hamb

...

nounExcept

indoda, amadoda

umZulu, amaZulu

Input/output

OWL/XML file

GUI output

Terminal output

[Keet et al.(2017)]
Algorithm 3 (AllSome) Verbalisation of “all-some” axiom type ($C \subseteq \exists R.D$)

Require: $C$ set of classes, language $L$ with $\sqsubseteq$ for subsumption and $\exists$ for existential quantification; variables: $A$ axiom, $NC_i$ noun class, $c_1, c_2 \in C$, $o \in R$, $a_1$ a term; $r_2, q_2$ concords; functions: $getFirstClass(A)$, $getSecondClass(A)$, $getNC(C)$, $getRC(NC_i)$, $getQC(NC_i)$, $getVSofOP(o)$.

Require: axiom $A$ with $\sqsubseteq$ has been retrieved and an $\exists$ on the rhs of the inclusion

1: $c_1 \leftarrow getFirstClass(A)$ \hspace{1cm} \{get subclass\}
2: $c_2 \leftarrow getSecondClass(A)$ \hspace{1cm} \{get superclass\}
3: $o \leftarrow getObjProp(A)$ \hspace{1cm} \{get object property\}
4: $v \leftarrow getVSofOP(o)$ \hspace{1cm} \{get verb stem of object property\}
5: $NC_1 \leftarrow getNC(c_1)$ \hspace{1cm} \{determine noun class by augment and prefix or dictionary\}
6: $NC_2 \leftarrow getNC(c_2)$ \hspace{1cm} \{determine noun class by augment and prefix or dictionary\}
7: $NC'_1 \leftarrow$ lookup plural nounclass of $NC_1$ \hspace{1cm} \{from known list\}
8: $c'_1 \leftarrow pluralise(c_1, NC'_1)$ \hspace{1cm} \{call algorithm pluralise to generate a plural from $o$\}
9: $a_1 \leftarrow$ lookup quantitative concord for $NC'_1$ \hspace{1cm} \{from quantitative concord (QC(all)) list\}
10: $r_2 \leftarrow getRC(NC_2)$ \hspace{1cm} \{get relative concord for $c_2$ from the QC$_{dwa}$ list\}
11: $q_2 \leftarrow getQC(NC_2)$ \hspace{1cm} \{get quantitative concord for $c_2$ from the QC$_{dwa}$ list\}
12: if $checkNegation(A) \equiv true$ then
13: \hspace{1cm} \{use negation (Algorithm 4)\}
14: else
15: \hspace{1cm} if $o$ annotated with present tense then
16: \hspace{1cm} \hspace{1cm} $connc1 \leftarrow$ lookup SC of $NC'_1$ \hspace{1cm} \{from known SC list\}
17: \hspace{1cm} \hspace{1cm} $o' \leftarrow connc1 v$ \hspace{1cm} \{generate conjugated verb\}
18: \hspace{1cm} $RESULT \leftarrow ‘a_1 c'_1 o’ a_2 r_2 q_2 dwa.’$ \hspace{1cm} \{verbalise the axiom\}
19: \hspace{1cm} else
20: \hspace{1cm} $RESULT \leftarrow ‘passive voice and inverses are not supported yet.’$
21: \hspace{1cm} end if
22: \hspace{1cm} end if
23: return $RESULT$

https://github.com/mkeet/GENIproject/
Proof-of-concept implementation (2/3)

```python
# simple existential quantification
# modified cf zulu rules 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':
        conjutr = sc_vowel_vroot(rt, ncp)
    else:
        sc = look_sc(ncp)
        conjutr = sc + rt
    return qca + ' ' + pl + ' ' + conjutr + 'a' + ' ' + super + ' ' + rc + qc + 'dwa'
```

https://github.com/mkeet/GENIproject
Proof-of-concept implementation (2/3)

```xml
<SubClassOf>
    <Class IRI="#indlovu"/>
    <Class IRI="#isilwane"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#indlovu"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#dla"/>
        <Class IRI="#ihlamvana"/>
    </ObjectSomeValuesFrom>
</SubClassOf>
```
Sentences outputted as pretty printing or plaintext (3/3)
Toward a proper and modular surface realiser

- MoReNL project: http://www.meteck.org/moreNL/
- Architecture design [Mahlaza and Keet(2022)] and development
- Proof-of-concept realiser for isiZulu and isiXosa: https://github.com/AdeebNqo/NguniTextGeneration (Zola Mahlaza)
- Model for template languages [Mahlaza and Keet(2021)]
- GUI for template creation
An ontology for template languages?

XML Schema elements

aligns to

Classes in the ontology that the disparate specifications are aligned to
An ontology for template language: ToCT

[Mahlaza and Keet(2021)]
Outline

1 Motivation
   • Context
   • Notes on NCB languages

2 Rule-based NLG
   • CNL and NLG in a nutshell
   • Generating basic sentences in isiZulu
   • Extending basic sentences

3 Discussion

4 Summary
Figuring out the present tense [Keet and Khumalo(2017b)]
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:
   \( post \rightarrow c \ | \ ca \ | \ car \ | \ cap \ | \ c \ r \ | \ crp \ | \ c \ p \ | \ arp \ | \ ar \ | \ arp \ | \ ap \ | \ r \ | \ rp \ | \ p \ | \ e \)
5. List of subject concords and negative subject concords (terminals for conjugation):
   \( s \rightarrow ngi \ | \ u \ | \ si \ | \ ni \ | \ ba \ | \ i \ | \ li \ | \ a \ | \ zi \ | \ lu \ | \ bu \ | \ ku \ | \ e \)
   \( ns \rightarrow angi \ | \ awu \ | \ aka \ | \ ali \ | \ asi \ | \ ayi \ | \ alu \ | \ abu \ | \ aku \ | \ ani \ | \ aba \ | \ awa \ | \ azi \ | \ e \)
6. List of mod:
   \( m \rightarrow a \ | \ e \ | \ ka \ | \ ma \ | \ nga \ | \ e \)
7. List of tense (present (e) and continuous (ya)tense; incomplete):
   \( t \rightarrow ya \ | \ e \)
8. List of aspect (additional rules omitted in this first iteration):
   \( asp \rightarrow sa \ | \ se \ | \ be \ | \ ile \ | \ e \)
9. List of object concords:
   \( o \rightarrow ngi \ | \ si \ | \ ku \ | \ ni \ | \ m \ | \ ba \ | \ wu \ | \ yi \ | \ li \ | \ wa \ | \ zi \ | \ lu \ | \ bu \ | \ e \)
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):
    \( ppre \rightarrow pl \ s \)
    \( pl \rightarrow aw \ | \ awu \ | \ mawu \ | \ e \ | \ ma \)
15. Static (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’;):
    \( wh \rightarrow ni \ | \ nini \ | \ phi \ | \ e \)
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:
    \( vr \rightarrow ab \ | \ . . . \ | \ zwib \)
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
Part-whole relations: main differences
[Keet and Khumalo(2020)]

- **part-of** (mereology)
- **s-part-of** (objects)
- **spatial-part-of**
- **contained-in** (3D objects)
- **located-in** (2D objects)
- **involved-in** (processes)
- **stuff-part-of** (different stuffs)
- **portion-of** (same stuff)
- **scattered-portion-of**
- **contiguous-portion-of**
- **member-of** (object/role-collective)
- **constitutes** (stuff-object)
- **participates-in** (object-process)
- **mpart-of** [in discourse only]

Part-whole relation
Part-whole relations: main differences
[Keet and Khumalo(2020)]
Part-whole relations: main differences
[Keet and Khumalo(2020)]

* includes, explicitly, also at least: involvement (between processes), participation of individual objects (cf. collectives) in events, and generic membership, in addition to those subsumed by *ingxenye* in this figure.
Extensions: part-whole relations

- ‘part’ *ingxenye* + ‘of’ <PC for NC of *ingxenye* that’s then phonologically conditioned with noun of the whole>
  - e.g.: ‘part of a human’
  - *ingxenye ya-* + *umuntu*
  - *ingxenye yomuntu*
Extensions: part-whole relations

- ‘part’ *ingxenye* + ‘of’ <PC for NC of *ingxenye* that’s then phonologically conditioned with noun of the whole>
  - e.g.: ‘part of a human’
    *ingxenye ya- + umuntu*
    *ingxenye yomuntu*

- ‘contained in’: locative affixes on the object that plays the container role
  - Each bolus of food is contained in some stomach
  - ‘bolus of food’ *indilinga yokudla* (nc9)
  - ‘stomach’ *isisu* (nc7)
  - ‘is contained in’: SC-EP-LOC-Whole-LOCSUF
    *zi-s-e-sis-wini* (phonological conditioning: e+i=e and u+ini=wini)
    *Zonke izindilinga zokudla zisesiswini esisodwa*
Outline

1 Motivation
   - Context
   - Notes on NCB languages

2 Rule-based NLG
   - CNL and NLG in a nutshell
   - Generating basic sentences in isiZulu
   - Extending basic sentences

3 Discussion

4 Summary
Further extensions and updates

- Adding (more) data-to-text to the knowledge-to-text
- Numbers, attributes (∼ adjectives), etc. etc.; e.g.:
  - engama-25 ([RC][COP][N prefix]-number)
  - Uqede imisebenzi eziyishumi ‘You completed ten exercises’, but amaphilisi ayishumi ‘ten pills’ [Keet(2021)]
  - izinkulungwane eziyisishiyagalombili namakhulu ayisishiyagalombili namashumi amane nane (numbers in speech cf. written)
- Option: application-driven prioritization for what to look into
- Rules-based approach is a slow process
- Limited documentation of language’s grammar, often outdated, incomplete, or incorrect
Initial results for other languages

- Multilingual pluraliser [Byamugisha et al.(2018)]
- Bootstrapping NLG for Runyankore [Byamugisha et al.(2016)]: it’s faster; (also shown by [Bosch et al.(2008)] for morphological analysers)
Initial results for other languages

- **Multilingual pluraliser** [Byamugisha et al.(2018)]
- **Bootstrapping NLG for Runyankore** [Byamugisha et al.(2016)]: it’s faster; (also shown by [Bosch et al.(2008)] for morphological analysers)
- **Bootstrappability strategies?**
  - Trying to understand morphological and verb similarities as proxies [Keet(2016), Mahlaza and Keet(2019)]
  - Guthrie zones (not a good predictor) [Byamugisha(2019)]
What about ML and such for NLG?

- Feasibility of using machine learning or deep learning for templates:
  - Lack of good and relevant data (e.g., bible and Ubuntu software manual are out-of-domain for healthcare messages, old texts, OCR errors and typos)
  - Need comparatively more data (recall agglutination and type-to-token ratio)
  - Needs good NLU algorithms
  - Computing the language models is computationally expensive
  - The systems “hallucinate” and have spurious repetitions, in English at least

- Jan Buys at UCT commenced with that approach

- Other efforts: mashakane (corpus & MT) and Qfrency (TTS)
Outline

1 Motivation
   • Context
   • Notes on NCB languages

2 Rule-based NLG
   • CNL and NLG in a nutshell
   • Generating basic sentences in isiZulu
   • Extending basic sentences

3 Discussion

4 Summary
Computational view on NCB languages wrt CNLs 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

Low resource languages a challenge for both rule-based and data-driven approaches, but in different ways; take your pick
Collaborators and Funding

- Main linguist: Langa Khumalo (SADiLaR)
- Current/former students wrt NLG and ontologies: Mary-Jane Antia, Joan Byamugisha, Catherine Chavula, Takunda Chirema, Leighton Dawson, Francis Gillis-Webber, Zola Mahlaza, Sindiso Mkhatshwa, Junior Moraba, Gerald Ngumbulu, Toky Raboanary, Musa Xakaza, Steve Wang

- Main NRF grants: GeNI & MoRENL projects
  http://www.meteck.org/files/geni/
  http://www.meteck.org/MoReNL/
Sonja Bosch, Laurette Pretorius, and Axel Fleisch.
Experimental bootstrapping of morphological analysers for nguni languages.

J. Byamugisha, C.M. Keet, and B. DeRenzi.
Bootstrapping a runyankore cnl from an isizulu cnl.

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.*

A. Gatt and E. Reiter.
Simplenlg: A realisation engine for practical applications.
F. Gillis-Webber, S. Tittel, and C. M. Keet.
A model for language annotations on the web.
24-28 June 2019, Villa Clara, Cuba.

Mustafa Jarrar, C. Maria Keet, and Paolo Dongilli.
Multilingual verbalization of ORM conceptual models and axiomatized ontologies.

C. M. Keet.
An assessment of orthographic similarity measures for several african languages.

C. M. Keet.
Representing and aligning similar relations: parts and wholes in isizulu vs english.

C. M. Keet and T. Chirema.
A model for verbalising relations with roles in multiple languages.
19-23 November 2016, Bologna, Italy.
C. M. Keet and L. Khumalo.
Toward a knowledge-to-text controlled natural language of isiZulu.

C. M. Keet and L. Khumalo.
Grammar rules for the isizulu complex verb.

C. M. Keet and L. Khumalo.
On the ontology of part-whole relations in Zulu language and culture.
17-21 September, 2018, Cape Town, South Africa.

C. Maria Keet.
Natural language generation requirements for social robots in subsaharan africa.
9-11 May, online.

C. Maria Keet and Langa Khumalo.
Toward verbalizing logical theories in isiZulu.
20-22 August 2014, Galway, Ireland.
C. Maria Keet and Langa Khumalo.
Basics for a grammar engine to verbalize logical theories in isiZulu.

C. Maria Keet and Langa Khumalo.
On the verbalization patterns of part-whole relations in isizulu.

C. Maria Keet and Langa Khumalo.
Parthood and part–whole relations in zulu language and culture.

C. Maria Keet, Musa Xakaza, and Langa Khumalo.
Verbalising owl ontologies in isizulu with python.

Tobias Kuhn.
A principled approach to grammars for controlled natural languages and predictive editors.

Z. Mahlaza and C. M. Keet.
A method for measuring verb similarity for two closely related languages with application to zulu and xhosa.
Z. Mahlaza and C. Maria Keet.

Z. Mahlaza and C. Maria Keet.

Z. Mahlaza and C.M. Keet.

E. Reiter and R. Dale.

K. Winkler, T. Kuhn, and M Volk.
Thank you!

Questions?

My award-winning textbook

A memoir
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\(^1\)

- What if your language doesn’t have an ISO language tag?
  - Create your own!
  - e.g., with MoLA [Gillis-Webber et al.(2019)]

- Multilingual ontologies vs multiple monolingual ontologies, management thereof

- (There are more engineering questions to make it work)

\(^1\)https://wikiworkshop.org/2022/papers/WikiWorkshop2022_paper_31.pdf