Tutorial: Generating text from ontologies in multiple languages

C. Maria Keet\textsuperscript{1} and Zola Mahlaza\textsuperscript{2}

\textsuperscript{1} University of Cape Town, South Africa, mkeet@cs.uct.ac.za
\textsuperscript{2} University of Pretoria, South Africa, z.mahlaza@up.ac.za

Joint Ontology Workshops 2022, 15-19 August 2022
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
Ontology verbalisation: The Idea

- Render the axioms from the ontology as a natural language sentence
**Ontology verbalisation: The Idea**

- Render the axioms from the ontology as a natural language sentence
  - Maybe sometimes a *pseudo*-natural language sentence (i.e., not 100% grammatically correct, but good enough)
Ontology verbalisation: The Idea

- Render the axioms from the ontology as a natural language sentence
  - Maybe sometimes a *pseudo*-natural language sentence (i.e., not 100% grammatically correct, but good enough)
- For some purpose; e.g.:
  - Make axioms accessible to non-logicians for validation
  - Generate answerable questions in some edTech tool
  - Input to an ontology-driven human-robot interaction module
- The kind of sentence. Mainly: statements or questions
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
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
Example: Query formulation with Quelo [7]
Basic idea of CNLs and NLG with ontologies
- The ‘ingredients’
- Variation in the components, with architectures
- Example systems

Hands-on exercises: build your own verbaliser (or part thereof)

Advanced topics
- Modelling effects
- ‘Localisation’ and multilingualism
- Research challenges
Outline

1 Introduction

2 Rule-based NLG
   • What is CNL, NLG?
   • Architectures
   • Evaluating NLG systems for ontologies

3 Hands-on

4 Advanced Topics
   • Modelling styles
   • Localisation and multilingualism

5 Summary
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
Short answer

- **Controlled Natural Language**: constrain the grammar or vocabulary (or both) of a natural language
- **Natural Language Generation**: 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 buqhubeke noma eyehlsa ubuhlungu?

**Dyspnoea**

Are you breathless at any time? - Uke uphelelewe umoya...
Ex: Avalanche bulletins with **canned segments** [23]

**Fig. 2.** Schema of a phrase in the source language German (above). \{on_steep\} mark a subsegment 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: crowd2.0 modelling tool (CDM & ontologies) [3]
Tutorial: Generating text from ontologies in multiple languages

- Rule-based NLG
- What is CNL, NLG?

Ex: crowd2.0 modelling tool (CDM & ontologies) [3]
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
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, generate different sentences; e.g.:
  - John eats an apple
  - He eats an apple
  - He eats it
  - John eats it
  ...

Ex.: Mixing grammar with templates
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 [17], Grammatical Framework (http://www.grammaticalframework.org/), SimpleNLG [8]**

See also: [18]
Tutorial: Generating text from ontologies in multiple languages

Rule-based NLG

What is CNL, NLG?

Business rules/conceptual data models and logic reconstruction

BR: Each Course is taught by at least one Professor

FOL: ∀x (Course(x) → ∃y (is_taught_by(x, y) ∧ Professor(y)))

DL: Course ⊑ ∃ is_taught_by.Professor

(i.e., a mandatory constraint / existential quantification)
Content stored in, e.g., some 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

Simple existential quantification (‘mandatory constraints’ in CDM parlance)

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

that and more for 11 languages (ORM terminology): [11]
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))))

((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))))

John eats it

She eats an apple
NL Grammars, illustration (1/2)

Sentence $\rightarrow$ NounPhrase $|$ VerbPhrase
NounPhrase $\rightarrow$ Adjective $|$ NounPhrase
NounPhrase $\rightarrow$ Noun

... 

Noun $\rightarrow$ car $|$ train
Adjective $\rightarrow$ big $|$ broken

... 

(and complexity of the grammar)

+ rules for verb tenses, pluralisation etc.
SimpleNLG tool [8] (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
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)
The ‘NLG pipeline’ of 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.)
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
Example tools

- Ontology $\rightarrow$ NL, NL $\rightarrow$ ontology, or both
- Subject domain independent ones: e.g., SWAT4NL, Rabbit, ACE, GF-based ones
- Tailored to the ontology: e.g., SNOMED CT verbaliser
- Review on ontology verbalisation: [22]
Example: the isiZulu verbaliser for ontologies (1/3)
Implementation (2/3)

```
450 <SubClassOf>
451   <Class IRI="#indlovu"/>
452   <Class IRI="#isilwane"/>
453 </SubClassOf>
454 <SubClassOf>
455   <Class IRI="#indlovu"/>
456   <ObjectSomeValuesFrom>
457     <ObjectProperty IRI="#dla"/>
458   </ObjectSomeValuesFrom>
459   <Class IRI="#ihlamvana"/>
460   </ObjectSomeValuesFrom>
461 </SubClassOf>
```

https://github.com/mkeet/GENIproject/
# 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)
    p1 = 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 + ' ' + p1 + ' ' + conjugrt + 'a' + ' ' + super + ' ' + rc + qc + 'dwa'

https://github.com/mkeet/GENIproject/
Sentences outputted as pretty printing or plaintext (3/3)
Choices and requirements (see also [2])

- Input

- Output

- Context
Choices and requirements (see also [2])

- **Input**
  - File type (OWL (species, syntax), CLIF, ....) and size
  - Subject domain- , task- dependent or not

- **Output**

- **Context**
Choices and requirements (see also [2])

- **Input**
  - File type (OWL (species, syntax), CLIF, ....) and size
  - Subject domain-, task-dependent or not

- **Output**
  - Sentences or paragraphs; in writing or TTS
  - Fluency of the text (syntactically, orthographically, grammatically correct or almost so)
  - Target language(s)

- **Context**
Choices and requirements (see also [2])

- **Input**
  - File type (OWL (species, syntax), CLIF, ....) and size
  - Subject domain- , task- dependent or not

- **Output**
  - Sentences or paragraphs; in writing or TTS
  - Fluency of the text (syntactically, orthographically, grammatically correct or almost so)
  - Target language(s)

- **Context**
  - Target genre and type of sentences, audience and user profile
  - Communicative goal
  - Setting of NL consumption (mobile, desktop, in museum etc)
What is there to configure, as a minimum? (1/2)

- One sentence per axiom type or more for variation
What is there to configure, as a minimum? (1/2)

- One sentence per axiom type or more for variation
- Template specification language

- Choice of whether to add some grammar; if so, how
What is there to configure, as a minimum? (1/2)

- One sentence per axiom type or more for variation
- Template specification language
  - Mostly done ad hoc (XML xsd, dtd)
  - Use an ontology for that for additional validation [20]
- Choice of whether to add some grammar; if so, how
What is there to configure, as a minimum? (1/2)

- One sentence per axiom type or more for variation
- Template specification language
  - Mostly done ad hoc (XML xsd, dtd)
  - Use an ontology for that for additional validation [20]
- Choice of whether to add some grammar; if so, how
  - Only ‘prettify’? e.g., “a(n)” vs indefArt(...), from “camelCase” to “camel case”
  - More extensive; e.g., referring expression generation, sentence aggregation
    “X produces Y.” + “X produces Z.” → “X produces Y and Z.”
- Grammar specification or engine
A note on template languages

- There are many
A note on template languages

- There are many
- Fully declarative ... functional
- How template components interact with the grammar (if allowed)
- The tasks of other modules in the realiser pipeline, if any
- Type of input it needs to process
- For which natural language
- Language to declare the templates in (XML, JSON, ...
Example: proposed template language for Abstract Wikipedia (summarised)

https://meta.wikimedia.org/wiki/Abstract_Wikipedia/Template_Language_for_Wikifunctions
An ontology for template languages?

XML Schema elements

<Object> uses types XML Schema elements

<Text> uses types

DOGMA's XML Templates

GF Templates uses types GF types

XXX Templates uses types XXX types

Word

Phrase

Slot

Classes in the ontology that the disparate specifications are aligned to

aligns to

uses types

uses types

uses types
An ontology for template language: ToCT [20]
What is there to configure, as a minimum? (2/2)

- The technology of the generation aspects:
  - Where the templates will come from

- Build your own system or reuse
What is there to configure, as a minimum? (2/2)

- The technology of the generation aspects:
  - Where the templates will come from
    - Hand-crafted from the start
    - Patterns from elsewhere
    - Mine text for sample sentence structures
  - Build your own system or reuse
What is there to configure, as a minimum? (2/2)

- The technology of the generation aspects:
  - Where the templates will come from
    - Hand-crafted from the start
    - Patterns from elsewhere
    - Mine text for sample sentence structures
  - Build your own system or reuse
    - Java with OWL API
    - Python with Owlready
    - ?? with ?? for ontology language X
    - Generic systems like SimpleNLG
Realisation architectures: some terminology [19]

- Structure selection: connects the surface realiser to prior modules; responsible for making linguistic decisions given the semantic input.
- Structure encoding: This is the method used to capture the sentential structure.
- Structure induction: This refers to the method used to create the structures used for capturing sentences.
- Structure linearisation: This is the formation of text from some ordering structure.
- Ranking: The candidate filtering mechanism is responsible for selecting one sentence/structure out of many candidate output sentences or sentential structures.
Realisation architecture categories

The modules labelled A and B represent different realisation types for structures.
The modules labelled A and B represent different combinations of structure selection, linearisation, and ranking.
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
How to evaluate?

- 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?
How to evaluate?

- 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?
- In times of ML/DL go fast & break things: BLEU & ROUGE (automated): similarity of output to training sentences
Evaluation – certain examples (1/3)

Which sentence verbalises Lion ⊑ = 4 hasPart.Leg best?

1. Each Lion hasPart exactly 4 Leg.
2. Each lion has part exactly 4 legs.
3. Each lion has as part exactly 4 legs.
4. Each lion has exactly 4 legs as part.
5. Each lion must have exactly 4 legs.
6. All lions have exactly 4 legs as part.
7. All lions have 4 legs.
8. A lion has 4 legs.
Evaluation – certain examples (2/3)

Judge the sentence “Each lion has as part exactly 4 legs.”

Syntax:
1. Correct
2. Minor flaw, but acceptable
3. Major flaw(s), unacceptable

Grammar and understandability:
1. Correct, understandable
2. Not incorrect but somewhat awkward or uncommon formulation, understandable
3. Incorrect, but understandable
4. Incorrect, incomprehensible
Evaluation – certain examples (3/3)

\((\text{DevM}^-)\) Mandatory dynamic evolution, past:

\[
o \in \text{DevM}^- \underset{C_1, C_2}{\xrightarrow{\mathcal{I}(t)}} (o \in C_1 \underset{t}{\xrightarrow{\mathcal{I}(t)}} \exists t' < t. o \in \text{Dev} \underset{C_1, C_2}{\xrightarrow{\mathcal{I}(t')}}).
\]

For instance, Butterfly and the Caterpillar it used to be.

a. Each ..C_1.. must have been a(n) ..C_2.., but is not a(n) ..C_2.. anymore.

b. Each ..C_1.. was a(n) ..C_2.. before, but is not a(n) ..C_2.. now.

c. If ..C_1.., then ..C_1.. was a(n) ..C_2.. before, but is not a(n) ..C_2.. anymore.

[12]
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
Hands-on

Tasks

- Main exercises at http: //www.meteck.org/MoReNL/JOW022handsonNLGonto.pdf
- Online verbalisers to explore:
  - Try out either one with your own ontology and one provided:
    ACE http://attempto.ifi.uzh.ch/site/docs/owl_to_ace.html
    or https://github.com/Kaljurand/owl-verbalizer
    or SWAT
    http://mcs.open.ac.uk/nlg/SWAT/Verbaliser.html
    with prolog or crowd2.0 https://crowd-app.fi.uncoma.edu.ar
Hands-on

Observations from the practical?

- Fetch name (BFO_0000015) vs Fetch label (BFO’s ‘process’)
- How many naming variants did you detect?
Observations from the practical?

- Fetch name (BFO_0000015) vs Fetch label (BFO’s ‘process’)
- How many naming variants did you detect?
- Some axiom types were easier to verbalise than others
### Inconvenient cases (1/2): cido-base.owl

**Description: COVID-19 disease process**

<table>
<thead>
<tr>
<th>Equivalent To</th>
</tr>
</thead>
<tbody>
<tr>
<td>'caused by infection with' some SARS-CoV-2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SubClass Of</th>
</tr>
</thead>
<tbody>
<tr>
<td>'caused by infection with' some SARS-CoV-2</td>
</tr>
<tr>
<td>'coronavirus infectious disease process'</td>
</tr>
<tr>
<td>BFO_0000055 some DOID_0080600</td>
</tr>
<tr>
<td>BFO_0000066 some NCBITaxon_33208</td>
</tr>
<tr>
<td>BFO_0000066 some UBERON_0002048</td>
</tr>
</tbody>
</table>

**General class axioms**

<table>
<thead>
<tr>
<th>SubClass Of (Anonymous Ancestor)</th>
</tr>
</thead>
<tbody>
<tr>
<td>'caused by infection with' some NCBITaxon_11118</td>
</tr>
<tr>
<td>'caused by infection with' some NCBITaxon_11118</td>
</tr>
</tbody>
</table>
Inconvenient cases (1/2): cido-base.owl

<table>
<thead>
<tr>
<th>Description: COVID-19 disease process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equivalent To</td>
</tr>
<tr>
<td>'caused by infection with' some SARS-CoV-2</td>
</tr>
<tr>
<td>SubClass Of</td>
</tr>
<tr>
<td>'caused by infection with' some SARS-CoV-2</td>
</tr>
<tr>
<td>'coronavirus infectious disease process'</td>
</tr>
<tr>
<td>BFO_0000055 some DOID_0080600</td>
</tr>
<tr>
<td>BFO_0000066 some NCBITaxon_33208</td>
</tr>
<tr>
<td>BFO_0000066 some UBERON_0002048</td>
</tr>
<tr>
<td>General class axioms</td>
</tr>
<tr>
<td>SubClass Of (Anonymous Ancestor)</td>
</tr>
<tr>
<td>'caused by infection with' some NCBITaxon_11118</td>
</tr>
<tr>
<td>'caused by infection with' some NCBITaxon_11118</td>
</tr>
</tbody>
</table>

- Can’t get a neat sentence out of that without additional processing
- Verbalise the equivalence axiom only? Asserted subclasses too? And the inherited ones?
Inconvenient cases (2/2): Stuff ontology

Each emulsion is equivalent to a dispersion colloid and has as part exactly 1 stuff that has as state a liquid state and the inverse of inheres in continuous medium and has as part exactly one stuff that has as state a liquid state and the inverse of inheres in dispersed phase.

Each emulsion is a dispersion colloid.

Each emulsion has exactly one stuff part that is in the liquid state that has the role of continuous medium.

Each emulsion has exactly one stuff part that is in the liquid state that has the role of dispersed phase.

Annotations: Emulsion

Annotations

rdfs:comment [language: en]
continuous medium: liquid
dispersed phase: liquid
Examples: milk, mayonnaise, hand cream
http://en.wikipedia.org/wiki/Colloid

Description: Emulsion

Equivalent To

DispersionColloid
and (hasPartStuff exactly 1 (Stuff
and (hasState some Liquid)
and (inverse (srlInheresIn) some ContinuousMedium))
and (hasPartStuff exactly 1 (Stuff
and (hasState some Liquid)
and (inverse (srlInheresIn) some DispersedPhase)))
Each emulsion is equivalent to a dispersion colloid and has as part exactly 1 stuff that has as state a liquid state and the inverse of inheres in continuous medium and has as part exactly one stuff that has as state a liquid state and the inverse of inheres in dispersed phase.
Inconvenient cases (2/2): Stuff ontology

- Each emulsion is equivalent to a dispersion colloid and has as part exactly 1 stuff that has as state a liquid state and the inverse of inheres in continuous medium and has as part exactly one stuff that has as state a liquid state and the inverse of inheres in dispersed phase.

- Each emulsion is a dispersion colloid. Each emulsion has exactly one stuff part that is in the liquid state that has the role of continuous medium. Each emulsion has exactly one stuff part that is in the liquid state that has the role of dispersed phase.
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
Sample scenario: Intelligent “Inquire biology” textbook

- Idea: annotate textbook with ontology, generate questions automatically, mark automatically
Sample scenario: Intelligent “Inquire biology” textbook

- Idea: annotate textbook with ontology, generate questions automatically, mark automatically

Figure: Annotated p132 of Biology (9th edition) by Campbell and Reece [4]:

1. Molecules have a type of energy called thermal energy (heat), due to their constant motion. One result of this motion is diffusion, the movement of molecules of any substance so that they spread out evenly into the available space. Each molecule moves randomly, yet diffusion of a population of molecules may be directional. To understand this process, let’s imagine a synthetic membrane separating pure water from a solution of a dye in water. Study Figure 7.13a carefully to appreciate how diffusion would result in both solutions having equal concentrations of the dye molecules. Once that point is reached, there will be a dynamic equilibrium, with as many dye molecules crossing the membrane from one side to the other.

2. concentration gradient: A region along which the density of a chemical substance increases or decreases.

3. Substance will diffuse from where it’s more concentrated to where it’s less concentrated.

What diffuses hydrated ions from hypertonic solutions?
What are the differences between active transport and diffusion?
What do transport proteins diffuse to hypertonic solutions?
What do transport proteins diffuse along/across cell poles?
NOTES QUESTIONS
Q: What is the relation between a carbohydrate and a biomembrane?

(example from and based on [4])
Intelligent “Inquire biology” textbook

Q: What is the relation between a carbohydrate and a biomembrane?

(example from and based on [4])
Q: What is the relation between a carbohydrate and a biomembrane?

A: A carbohydrate side chain is a carbohydrate that is part of a glycolipid that is part of a biomembrane

(example from and based on [4])
Try to do that for *any* ontology [21]

- Question templates for different types of educational questions
  
  *Is a* `<T_NOUN>` `<OP_IS_PARTICIPLE_BY>` `<Quantifier_some>` `<T_NOUN>`?  
  
  *What does a* `<T_NOUN>` `<OP_VERB>`?
Try to do that for any ontology [21]

- Question templates for different types of educational questions
  
  *Is a \(<T\_NOUN>\) \(<OP\_IS\_PARTICIPLE\_BY>\) \(<Quantifier\_some>\) \(<T\_NOUN>\)?*

  *What does a \(<T\_NOUN>\) \(<OP\_VERB>\)?*

- Notion of “axiom prerequisites”

- (Assumes a certain way of representing something and of verbalising something)
Sample educational questions

Q: Is Monoclonal antibody reagent produced by some Hybridoma cell line?
A: Yes

Q: What does a Hybridoma cell line produce?
A: Monoclonal antibody reagent
Sample educational questions

Q: Is Monoclonal antibody reagent produced by some Hybridoma cell line?
A: Yes

Q: What does a Hybridoma cell line produce?
A: Monoclonal antibody reagent

Algorithm TBD, for every ontology pattern
Sample educational questions

Q: Is Monoclonal antibody reagent produced by some Hybridoma cell line?
A: Yes

Q: What does a Hybridoma cell line produce?
A: Monoclonal antibody reagent

Q: What does a Hybridoma cell line participate in?
A: Production

Algorithm TBD, for every ontology pattern.
Sample educational questions

Q: Is Monoclonal antibody reagent produced by some Hybridoma cell line?
A: Yes

Q: What does a Hybridoma cell line produce?
A: Monoclonal antibody reagent

Q: What does a Hybridoma cell line participate in?
A: Production

Algorithm TBD, i.e. every ontology pattern...
Sample educational questions

Q: Is Monoclonal antibody reagent produced by some Hybridoma cell line?
A: Yes

Q: What does a Hybridoma cell line produce?
A: Monoclonal antibody reagent

Q: What does a Hybridoma cell line participate in?
A: Production of monoclonal antibody reagents
Ontology patterns: to reify or not to reify... [6]
Correct but awkward or confusing sentences

(BioTop) Does a material object project onto an immaterial three dimensional physical entity?

(BioTop) A taxon quality projects onto a taxon value region. True or false?

note: taxon value region is an “abstract region in which the values of biological taxa are located (cf. Schulz et.al ISMB 2008).”
Correct but awkward or confusing sentences

(BioTop) Does a material object project onto an immaterial three dimensional physical entity?

(BioTop) A taxon quality projects onto a taxon value region. True or false?

note: taxon value region is an “abstract region in which the values of biological taxa are located (cf. Schulz et.al ISMB 2008).”

(SO) Does a mixed stuff have a part stuff that is a stuff?

MixedStuff ≡ Stuff ⊓ ∃hasPartStuff.Stuff
Correct but awkward or confusing sentences

(BioTop) Does a material object project onto an immaterial three dimensional physical entity?

(BioTop) A taxon quality projects onto a taxon value region. True or false?

note: taxon value region is an “abstract region in which the values of biological taxa are located (cf. Schulz et.al ISMB 2008).”

(SO) Does a mixed stuff have a part stuff that is a stuff?

\[
\text{MixedStuff} \equiv \text{Stuff} \sqcap \exists \text{hasPartStuff}.\text{Stuff}
\]

(BioTop) Which condition has a life that is some life?

\[
\text{condition} \equiv \text{disposition} \sqcup \text{function} \sqcup \text{material object} \sqcup \text{process}
\]

material object $\sqsubseteq \exists \text{has life}.\text{life}$
Better results with certain domain ontologies

- An applied style
- With abundance in granularity of relations
- Not just/mainly a ‘bare’ hierarchy, not just domain & range axioms
- ‘hasX’ and other OP naming considerations
- Naming in ontology often within-context, but out-of-context in the self-standing questions
Ontology patterns and styles

- Which patterns are there
- If the neatly combine, ontology ’styles’ emerge
- Either swap styles or modify algorithms
  - (More about that in my bio-ontologies keynote at ISMB’22)
Outline

1. Introduction

2. Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3. Hands-on

4. Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5. Summary
A first hurdle

- herbivoro
  - damanRoquero
  - elefante
  - impala
  - jabali
  - jirafa

- isilwane
  - ibhubesi
  - ihebhivo
    - indlovu
    - ikanivvo
    - impala

Description: herbivoro

Equivalent To

- (come only planta) or (come only (es parte de some planta))

Description: ihebhivo

Equivalent To

- (dla only isitshalo) or (dla only (ingxenye some isitshalo))
Tutorial: Generating text from ontologies in multiple languages

Advanced Topics

Localisation and multilingualism

Tool localisation?

NLG in languages other than English – more grammar

- Cases and gender in German, noun classes in isiZulu
- ‘Deep prepositions’ in isiZulu, Lithuanian
- Extensive phonological conditioning, agglutination, inflection, etc.
NLG in languages other than English – more grammar

- Cases and gender in German, noun classes in isiZulu
- ‘Deep prepositions’ in isiZulu, Lithuanian
- Extensive phonological conditioning, agglutination, inflection, etc.
  ⇒ Can’t ‘just fetch and tweak’ vocabulary
Examples: German

- Employee ⊑ ∃ worksFor . Company
  
arbeitet für ‘works for’ with 3rd case / dativ
  
der Betrieb ‘company’, M; die Firma, F
- Alle Angestellten arbeiten für mindestens einem Betrieb
- Alle Angestellten arbeiten für mindestens einer Firma
Examples: isiZulu

Axiom type \( \forall x(\text{X}(x) \rightarrow \exists y(\text{R}(x, y) \land \text{Y}(y))) \)
\[ X \subseteq \exists R.Y \]

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
\[ \text{QC}(\text{X}(x), y) \text{RC}_{\text{nc}} \text{-SC}_{\text{nc}} \text{-} \text{dwa} \]
Bonke oSolwazi bafundisa isifundo esisodwa
Bonke oSolwazi babhala incwadi eyodwa
Onke amakhanivo adla isilwane esisodwa
Zonke izindlovu zidla i-apula elilodwa

\[ \forall x(\text{uSolwazi}(x) \rightarrow \exists y(\text{fundisa}(x, y) \land \text{isifundo}(y))) \]
\[ \text{uSolwazi} \subseteq \exists \text{fundisa.isifundo} \]
Possible consequences and design questions

- How to deal with the vocabulary?

- Where to store the grammatical features and the processing?

- How to connect that extra baggage?

- Where does it affect the verbalisation architecture?
Possible consequences and design questions

- How to deal with the vocabulary?
  - Monolingual ontology or multilingual ontology
- Where to store the grammatical features and the processing?
- How to connect that extra baggage?
- Where does it affect the verbalisation architecture?
Possible consequences and design questions

- How to deal with the vocabulary?
  - Monolingual ontology or multilingual ontology

- Where to store the grammatical features and the processing?
  - ad hoc in same file after the template spec
  - functions embedded in the templates
  - separate grammar engine

- How to connect that extra baggage?

- Where does it affect the verbalisation architecture?
Possible consequences and design questions

- How to deal with the vocabulary?
  - Monolingual ontology or multilingual ontology

- Where to store the grammatical features and the processing?
  - ad hoc in same file after the template spec
  - functions embedded in the templates
  - separate grammar engine

- How to connect that extra baggage?
  - to templates
  - to ontology vocabulary in the annotation field
  - in a separate language model

- Where does it affect the verbalisation architecture?
Ontolex-Lemon

W3C community standard “to provide rich linguistic grounding for ontologies”

https://www.w3.org/2016/05/ontolex/
Sample entry linking an OWL class to language information

Link the word elephant to the entity in the ontology:

:lex_elephant a ontolex:LexicalEntry;
    ontolex:canonicalForm :form_elephant;

And have it in the lexicon:

:form_elephant a ontolex:Form;
    ontolex:writtenRep "elephant"@en.

For multiple languages: add language (from ISO639), add sense, assert translation
https://www.w3.org/2016/05/ontolex/#translation
More detailed aspects: complications with OPs [13]

- That *für* with dative on the article of the noun of the entity that plays the object role
- The ‘of’ in ‘part of’, in isiZulu: attached to the noun of the entity that plays the role of whole
- *umuntu uda* but *inja idla* (human vs dog eats): verb conjugation depends on the noun class of the noun of the entity that plays the subject role
More detailed aspects: prepositions [13]

```
{ For each ordered for,
  the Roles ordered for a Predicate are contained in a Relationship that is the Relationship of that Predicate.
  (but a Relationship that contains Roles need not have a Predicate, and Roles need not be ordered for a Predicate)

Each Axiom has participant either a Relationship or an Entity type or both, or an n-ary Predicate or an Entity type or both. }
```
More detailed aspects: prepositions [13]
More linguistics for annotations?

Example: GOLD ontology [5]. Selection:
Two key options for the system

1. Source language → English → verbalise → translate to source language
   E.g.: use grammatical framework or some translator
Two key options for the system

1. **Source language → English → verbalise → translate to source language**
   
   E.g.: use grammatical framework or some translator

2. **de novo, with language-specific additional data structures and rules**
   
   Can reuse idea of pipeline modules
   
   Can reuse model for multilingual labels and attendant models (e.g., ontolex-lemon, [13], MoLA [9])
Example of a ‘detour’ approach [10]

Depiction of the data flow that includes the automatic translations among the Latvian and English CNLs and OWL.

Tooling: ACE-OWL (to/from OWL) and the GF grammars for Latvian↔English.
Example of ‘reuse’ option [1]

NLG pipeline – with common modules – for generating museum text (in Greek) from ontologies.
Example of ‘for any language’ option (perhaps)

https://meta.wikimedia.org/wiki/Abstract_Wikipedia/Natural_language_generation_system_architecture_proposal (image by By Ariel Gutman)
Outline

1 Introduction

2 Rule-based NLG
   - What is CNL, NLG?
   - Architectures
   - Evaluating NLG systems for ontologies

3 Hands-on

4 Advanced Topics
   - Modelling styles
   - Localisation and multilingualism

5 Summary
Summary

- Core design choices for CNLs & NLG for ontologies
- Recurring hurdles
- Modelling patterns and styles affecting verbalisation
- Languages other than English, multilingualism
- A few applications and application scenarios
Collaborators and Funding

- Pablo Fillottrani, Ariel Gutman, Langa Khumalo
- 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

- NRF grant (for this tutorial): MoRENL project
http://www.meteck.org/MoReNL/
References I

I. Androutsopoulos, G. Lampouras, and D. Galanis.
Generating natural language descriptions from owl ontologies: the naturalowl system.

N. Bouayad-Agha, G. Casamayor, and L. Wanner.
Natural language generation in the context of the semantic web.

G. Braun, P. R. Fillottrani, and C. M. Keet.
A framework for interoperability between models with hybrid tools.

Inquire biology: A textbook that answers questions.

S. Farrar and D. T. Langendoen.
A linguistic ontology for the semantic web.

Pablo R. Fillottrani and C. Maria Keet.
Patterns for heterogeneous tbox mappings to bridge different modelling decisions.
In E. Blomqvist et al., editors, *Proceeding of the 14th Extended Semantic Web Conference (ESWC’17)*,
30 May - 1 June 2017, Portoroz, Slovenia.
Enrico Franconi, Paolo Guagliardo, and Marco Trevisan.
An intelligent query interface based on ontology navigation.
Hong Kong, February 2010.

A. Gatt and E. Reiter.
Simplenlg: A realisation engine for practical applications.
March 30-31, 2009, Athens, Greece.

F. Gillis-Webber, S. Tittel, and C. M. Keet.
A model for language annotations on the web.
24-28 June 2019, Villa Clara, Cuba.

Normunds Gruzitis and Guntis Barzdins.
Towards a more natural multilingual controlled language interface to OWL.

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

C. M. Keet.
Natural language template selection for temporal constraints.
21-23 September 2017, Bolzano, Italy.

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

C. Maria Keet and Langa Khumalo.
Toward verbalizing logical theories in isiZulu.

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

Z. Mahlaza and C. Maria Keet.
Formalisation and classification of grammar and template-mediated techniques to model and ontology verbalisation.

Z. Mahlaza and C. Maria Keet.
Surface realisation architecture for low-resourced african languages.

Z. Mahlaza and C.M. Keet.
Toct: A task ontology to manage complex templates.
T. Raboanary, S. Wang, and C. M. Keet.
Generating answerable questions from ontologies for educational exercises.
29 Nov - 3 Dec 2021, online.

Hazem Safwat and Brian Davis.
CNLs for the semantic web: a state of the art.

K. Winkler, T. Kuhn, and M Volk.
Evaluating the fully automatic multi-language translation of the swiss avalanche bulletin.
20-22 August 2014, Galway, Ireland.
Thank you!

Questions?

My award-winning textbook

A memoir
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 [15, 16, 14]
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)


### 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 $\cap$, 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
  - $\neg C$, $C \cap D$, and $C \sqcup D$ are concepts, and
  - $\forall R.C$ and $\exists R.C$ are concepts
- **Individuals**
- e.g., $\text{Lion} \sqsubseteq \exists \text{eats.Herbivore} \sqcap \forall \text{eats.Herbivore}$
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 ⊑

\[(U1)\] 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)
### Summary

Generating basic sentences in isiZulu

<table>
<thead>
<tr>
<th>NC</th>
<th>QC (all)</th>
<th>NEG SC</th>
<th>PRON</th>
<th>RC</th>
<th>QC$_{dwa}$</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>u-onke $\rightarrow$ wonke</td>
<td>wo-</td>
<td>aka-</td>
<td>yena</td>
<td>o-</td>
<td>ye-</td>
</tr>
<tr>
<td>2</td>
<td>ba-onke $\rightarrow$ bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
</tr>
<tr>
<td>1a</td>
<td>u-onke $\rightarrow$ wonke</td>
<td>wo-</td>
<td>aka-</td>
<td>yena</td>
<td>o-</td>
<td>ye-</td>
</tr>
<tr>
<td>2a</td>
<td>ba-onke $\rightarrow$ bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
</tr>
<tr>
<td>3</td>
<td>u-onke $\rightarrow$ wonke</td>
<td>wo-</td>
<td>aka-</td>
<td>wona</td>
<td>o-</td>
<td>ye-</td>
</tr>
<tr>
<td>2a</td>
<td>ba-onke $\rightarrow$ bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>aba-</td>
<td>bo-</td>
</tr>
<tr>
<td>3</td>
<td>u-onke $\rightarrow$ wonke</td>
<td>wo-</td>
<td>awu-</td>
<td>wona</td>
<td>o-</td>
<td>wo-</td>
</tr>
<tr>
<td>4</td>
<td>i-onke $\rightarrow$ yonke</td>
<td>yo-</td>
<td>ayi-</td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
</tr>
<tr>
<td>5</td>
<td>li-onke $\rightarrow$ lonke</td>
<td>lo-</td>
<td>ali-</td>
<td>lona</td>
<td>eli-</td>
<td>lo-</td>
</tr>
<tr>
<td>6</td>
<td>a-onke $\rightarrow$ onke</td>
<td>o-</td>
<td>awa-</td>
<td>wona</td>
<td>a-</td>
<td>wo-</td>
</tr>
<tr>
<td>7</td>
<td>si-onke $\rightarrow$ sonke</td>
<td>so-</td>
<td>asi-</td>
<td>sona</td>
<td>esi-</td>
<td>so-</td>
</tr>
<tr>
<td>8</td>
<td>zi-onke $\rightarrow$ zonke</td>
<td>zo-</td>
<td>azi-</td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
</tr>
<tr>
<td>9a</td>
<td>i-onke $\rightarrow$ yonke</td>
<td>yo-</td>
<td>ayi-</td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
</tr>
<tr>
<td>(6)</td>
<td>a-onke $\rightarrow$ onke</td>
<td>o-</td>
<td>awa-</td>
<td>wona</td>
<td>a-</td>
<td>wo-</td>
</tr>
<tr>
<td>9</td>
<td>i-onke $\rightarrow$ yonke</td>
<td>yo-</td>
<td>ayi-</td>
<td>yona</td>
<td>e-</td>
<td>yo-</td>
</tr>
<tr>
<td>10</td>
<td>zi-onke $\rightarrow$ zonke</td>
<td>zo-</td>
<td>azi-</td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
</tr>
<tr>
<td>11</td>
<td>lu-onke $\rightarrow$ lonke</td>
<td>lo-</td>
<td>alu-</td>
<td>lona</td>
<td>olu-</td>
<td>lo-</td>
</tr>
<tr>
<td>(10)</td>
<td>zi-onke $\rightarrow$ zonke</td>
<td>zo-</td>
<td>azi-</td>
<td>zona</td>
<td>ezi-</td>
<td>zo-</td>
</tr>
<tr>
<td>14</td>
<td>ba-onke $\rightarrow$ bonke</td>
<td>bo-</td>
<td>abu-</td>
<td>bona</td>
<td>obu-</td>
<td>bo-</td>
</tr>
<tr>
<td>15</td>
<td>ku-onke $\rightarrow$ konke</td>
<td>zo-</td>
<td>aku-</td>
<td>khona</td>
<td>oku-</td>
<td>zo-</td>
</tr>
</tbody>
</table>
## Summary

Generating basic sentences in isiZulu

<table>
<thead>
<tr>
<th>NC</th>
<th>QC&lt;sub&gt;oral&lt;/sub&gt; → wonke</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>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>bona</td>
<td>ab-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>1a</td>
<td>u-onke → 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 → bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>ab-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>3a</td>
<td>u-onke → 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 → bonke</td>
<td>bo-</td>
<td>aba-</td>
<td>bona</td>
<td>ab-</td>
<td>bo-</td>
<td>ba-</td>
</tr>
<tr>
<td>3</td>
<td>u-onke → 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 → 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 → 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 → 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 → 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 → 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 → 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 → 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 → 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 → 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 → 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 → 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 → 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 → konke</td>
<td>zo-</td>
<td>aku-</td>
<td>khona</td>
<td>oku-</td>
<td>zo-</td>
<td>ku-</td>
</tr>
</tbody>
</table>
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 ⊑ 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. $N_1 \ <\text{copulative } ng/y\text{ depending on first letter of } N_2 > N_2$.  
b. $<\text{plural of } N_1 > \ <\text{copulative } ng/y\text{ depending on first letter of plural of } N_2 > <\text{plural of } N_2 >$.  
c. $<\text{All-concord for NC}_x > + \text{onke } <\text{plural of } N_1 , \text{being of } NC_x > <\text{copulative } ng/y\text{ depending on first letter of } N_2 > N_2$. 
Existential Quantification

(E1) Giraffe [Sub] [∃] 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

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

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

\( u\text{Solwazi} \subseteq \exists -\text{fundisa}.\text{Isifundo} \)
Generating basic sentences in isiZulu

\[ \forall x \ (u\text{Solwazi}(x) \rightarrow \exists f \text{-func}) \]

<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>ma-</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>ma-</td>
</tr>
<tr>
<td>(6)</td>
<td>a-</td>
<td>ma-</td>
</tr>
<tr>
<td>9</td>
<td>i(n)-</td>
<td>m(u)-</td>
</tr>
<tr>
<td>10</td>
<td>i-</td>
<td>(li)-</td>
</tr>
<tr>
<td>11</td>
<td>u-</td>
<td>lu-</td>
</tr>
<tr>
<td>(10)</td>
<td>i-</td>
<td>(li(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>QCoral+onke</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>1a</td>
<td>u-onke → wonke</td>
</tr>
<tr>
<td>2a</td>
<td>ba-onke → bonke</td>
</tr>
<tr>
<td>3a</td>
<td>u-onke → wonke</td>
</tr>
<tr>
<td>(2a)</td>
<td>ba-onke → bonke</td>
</tr>
<tr>
<td>3</td>
<td>u-onke → wonke</td>
</tr>
<tr>
<td>4</td>
<td>i-onke → yonke</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>9</td>
<td>i-onke → yonke</td>
</tr>
<tr>
<td>10</td>
<td>zi-onke → zonke</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>

Bonke oSolwazi
Tutorial: Generating text from ontologies in multiple languages

Summary

Generating basic sentences in isiZulu

∀x (uSolwazi(x) → ∃y (fumilisa(x, y) ∧ Isifundo(y))))

Bonke oSolwazi bafundisa

reuse pluralised
NC of subject
look-up SC
of that NC

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

\[ u\text{Solwazi} \sqsubseteq \exists \ -\text{fundisa. Isifundo} \]

Bonke oSolwazi bafundisa Isifundo
∀x (uSolwazi(x) → ∃y uSolwazi ∈ fundisa.I)

Bonke oSolwazi bafundisa Isifundo esisodwa