Toward isiZulu Natural Language Generation

C. Maria Keet[1]

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

CS Colloquium @UCT, 12 June 2014

[1] Joint work with Dr. Langa Khumalo, Linguistics program and Director of the University Language Planning and Development Office, University of KwaZulu-Natal

Outline

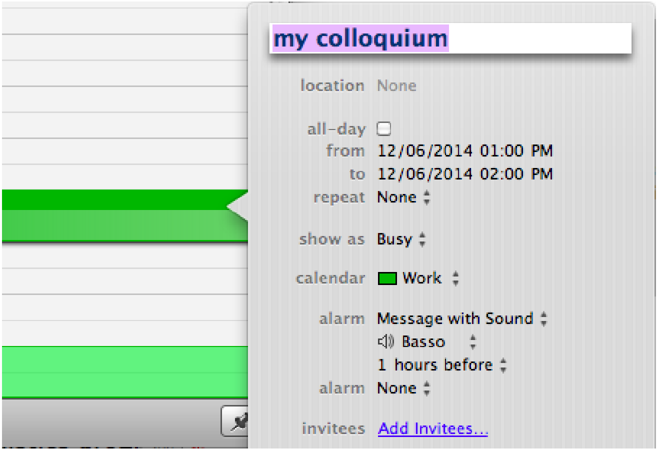
Outline

1. Motivation
   - A few application scenarios
   - NLG and knowledge management
2. isiZulu intro
3. isiZulu NLG
   - Patterns and options
   - Survey results
   - Algorithms for selected constructs
4. Discussion
5. Conclusions

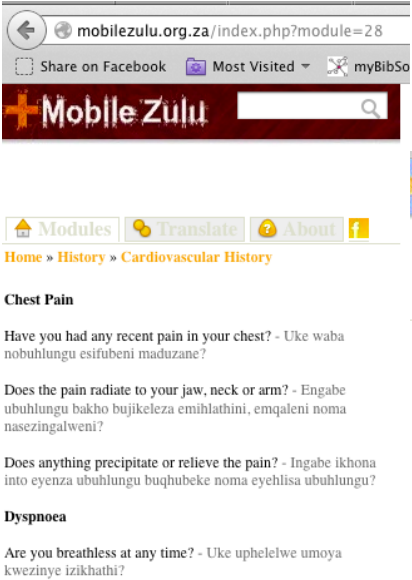
Natural language interfaces with some NLG

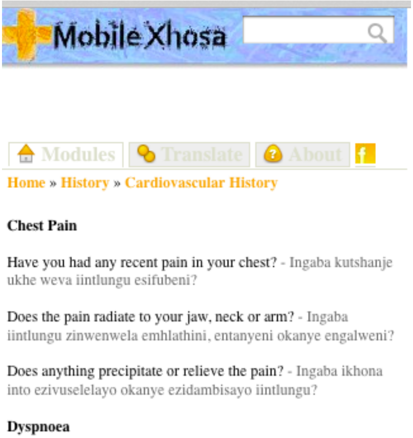
- Many tools, webpages, etc. with some natural language component
- Querying of information in natural language (cf. a query language SQL, SPARQL)
- Business rules typically specified in a natural language
- etc.

Example: iCal calendar entry with canned text

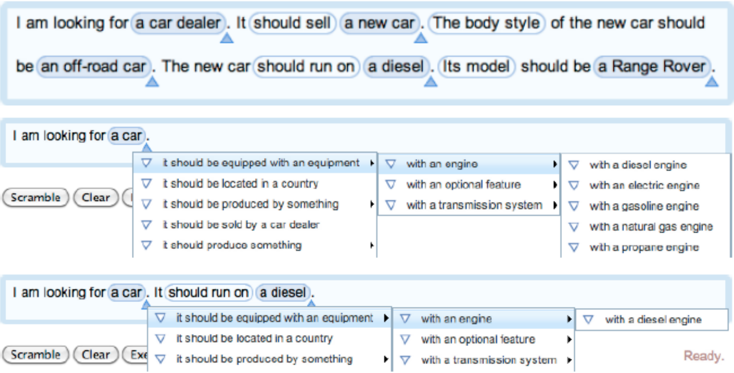


Example: Saadiq Moolla's mobile healthcare app





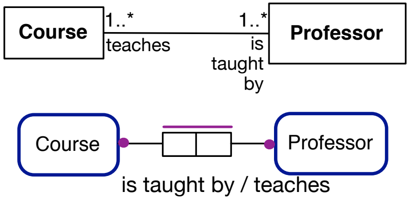
Example: Query formulation with Quelo [Franconi et al.(2010)]



Pictures from: Quelo @ The IESD Challenge 2012

Demo at: `http://krdbapp.inf.unibz.it:8080/quelo/`

Example: Business rules and conceptual data models



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

Each Professor teaches **at least one** Course

NLG, principal approaches

- Canned text
- Templates
  - Notably for English [Fuchs et al.(2010), Schwitter et al.(2008), Third et al.(2011), Curland and Halpin(2007)],
  - but also other languages [Jarrar et al.(2006)]
- Grammar engines, such as [Kuhn(2013)], Grammatical Framework (`http://www.grammaticalframework.org/`)

⇒ Controlled Natural Language

NLG, principal approaches

- Canned text
- Templates
  - Notably for English [Fuchs et al.(2010), Schwitter et al.(2008), Third et al.(2011), Curland and Halpin(2007)],
  - but also other languages [Jarrar et al.(2006)]
- Grammar engines, such as [Kuhn(2013)], Grammatical Framework (http://www.grammaticalframework.org/)

⇒ Controlled Natural Language

Business rules/conceptual data models and logic reconstruction

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

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

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

Example of templates

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

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

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

Example of templates

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

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

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

Example of templates

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

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

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

Example of templates

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

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

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

NL Grammars, illustration

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

(and complexity of the grammar)

Question

- Can the template-based approach be used also for isiZulu NLG?
  - If so, create those templates
  - If not, start with basics for a grammar engine
- Use a practically useful language to benefit both ICT and linguists and, possibly, some subject domain (e.g., medicine, NRS [Alberts et al.(2012)])
- Details in [Keet and Khumalo(2014b), Keet and Khumalo(2014a)]

Question

- Can the template-based approach be used also for isiZulu NLG?
  - If so, create those templates
  - If not, start with basics for a grammar engine
- Use a practically useful language to benefit both ICT and linguists and, possibly, some subject domain (e.g., medicine, NRS [Alberts et al.(2012)])
- Details in [Keet and Khumalo(2014b), Keet and Khumalo(2014a)]

Question

- Can the template-based approach be used also for isiZulu NLG?
  - If so, create those templates
  - If not, start with basics for a grammar engine
- Use a practically useful language to benefit both ICT and linguists and, possibly, some subject domain (e.g., medicine, NRS [Alberts et al.(2012)])
- Details in [Keet and Khumalo(2014b), Keet and Khumalo(2014a)]

Outline

A few features of isiZulu

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

Abafana abancane bazozithenga izincwadi ezinkulu

aba-fana	**aba**-ncane	**ba**-	zo-	**zi**- thenga	**izi**-ncwadi	e-**zi**-nkulu
2.boy	**2**.small	**2.SUBJ**-FUT-**10.OBJ**-buy			**10**.book	REL-**10**.big

'The little boys will buy the big books'

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

Outline

Logic foundation for isiZulu NLG

- Roughly OWL 2 EL
- OWL 2 EL is a W3C-standardised profile of OWL 2
- Tools, ontologies in OWL 2 (notably SNOMED CT)
- On the ‘roughly’: minus transitivity, but with negation, amounting to $\mathcal{ALC}$
  - of that, we have patterns for universal and existential quantification, subsumption, negation (disjointness), and conjunction
  - union not yet covered explicitly, but note $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$
  - more detail on the languages: see the Description Logics Handbook [Baader et al.(2008)] and OWL 2 Standard

Logic foundation for isiZulu NLG

- Roughly OWL 2 EL
- OWL 2 EL is a W3C-standardised profile of OWL 2
- Tools, ontologies in OWL 2 (notably SNOMED CT)
- On the 'roughly': minus transitivity, but with negation, amounting to $\mathcal{ALC}$
  - of that, we have patterns for universal and existential quantification, subsumption, negation (disjointness), and conjunction
  - union not yet covered explicitly, but note $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$
  - more detail on the languages: see the Description Logics Handbook [Baader et al.(2008)] and OWL 2 Standard

$\mathcal{ALC}$ syntax

- *Concepts* denoting entity types/classes/unary predicates/universals, including top $\top$ and bottom $\bot$;
- *Roles* denoting relationships/associations/n-ary predicates/properties;
- *Constructors*: and $\sqcap$, or $\sqcup$, and not $\neg$; quantifications forall $\forall$ and 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*

$\mathcal{ALC}$ semantics

- *domain of interpretation*, and an *interpretation*, where:
  - Domain Δ is a non-empty set of objects
  - Interpretation: $\cdot^{\mathcal{I}}$ is the *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 $\bot^{\mathcal{I}} = \emptyset$
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y. R^{\mathcal{I}}(x,y) \rightarrow C^{\mathcal{I}}(y)\}$
- $(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y. R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y)\}$

A few constructors, their typical verbalization in English, and the basic options in isiZulu

DL symbol	Sample verbalization English	Sample verbalization in isiZulu (see text for additional rules)
$\sqsubseteq$	... is a ...	Depends on what is on the rhs of $\sqsubseteq$ and desideratum: A) semantic distinction i) yi/ongu/uyi/ngu (living thing) ii) iyi (non-living thing) B) syntactic distinction iii) ng (nouns commencing with a, o, or u) iv) y (nouns commencing with i)
$\sqcap$	... and ...	Depends on the use of the $\sqcap$: i) ... na/ne/no ... (list of things) ii) 1) ... futhi ... (connective) 2) ... kanye ... (connective)
$\neg$	not ...	angi/akusiso/akusona/akubona/akulona/asibona/ akalona/akuyona
$\exists$	1) some ... 2) there exists ... 3) at least one ...	Depends on position in axiom: I. quantified over class, depends on meaning of class: i) kuno (living thing) ii) kune (non-living thing) II. includes relation (preposition issue omitted): 1) ... [concords]dwa 2) ... noma [copulative + concord]phi ... 3) thize
$\forall$	1) for all ... 2) each ...	Depends on what it is quantified over: A) semantic distinction i) wonke/bonke/sonke/zonke (living thing) ii) onke/konke/lonke/yonke (non-living thing) B) another semantic distinction i) use noun class (see Table 8)

Universal Quantification

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

(U1) Boy $\sqsubseteq$ ...

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

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

(U2) Phone $\sqsubseteq$ ...

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

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

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

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

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

Subsumption

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

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

ikhambi ngumuthi ('medicinal herb is a plant')

amakhambi yimithi ('medicinal herbs are plants')

wonke amakhambi ngumuthi ('all medicinal herbs are a plant')

(S2) `Giraffes` $\sqsubseteq$ `Animals`

izindlulamithi yizilwane ('giraffes are animals'; generic)

(S3) `Cellphone` $\sqsubseteq$ `Phone`

Umakhalekhukhwini uyifoni ('cellphone is a phone'; determ.)

Possible subsumption patterns

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

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

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

Subsumption: adding negation

- Need to choose between
  - singular and plural, and with or without the universal quantification voiced
- Copulative is omitted
- Combines the negative subject concord (NEG SC) of the noun class of the first noun (*aku-*) with the pronomial (PRON) of the noun class of second noun (*-yona*)

(SN1) `Cup` $\sqsubseteq$ `¬Glass`

indebe akuyona ingilazi ('cup not a glass')

zonke izindebe aziyona ingilazi ('all cups not a glass')

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

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

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

Possible negation (disjointness) patterns

a. <N_1 of NC_x> <NEG SC of NC_x><PRON of NC_y> <N_2 of NC_y>.

b. <All-concord for NC_x>onke <plural N_1, being of NC_x> <NEG SC of NC_x><PRON of NC_y> <N_2 with NC_y>.

Conjunction

- Conjunction as enumeration uses *na*
- Changes into (a + i =) *ne* or (a + u =) *no*, depending on the first letter of the second noun
- Prefixed to the second noun that drops its first letter
- Conjunction as connective of clauses: *kanye* or *futhi*

(C1) $\texttt{Milk} \sqcap \texttt{Butter}$

Ubisi <u>ne</u>bhotela (*Ubisi* + *na* + *Ibhotela*)

(C2) $\texttt{Butter} \sqcap \texttt{Milk}$

Ibhotela <u>no</u>bisi (*Ibhotela* + *na* + *Ubisi*)

(C3) $\ldots \exists \texttt{has\_filling.Cream} \sqcap \exists \texttt{has\_Icing.Lemon\_flavour} \ldots$

...kune zigcwalisa ukhilimu <u>kanye</u> nezinye uqweqwe olunambitheka_ulamula...

...kune zigcwalisa ukhilimu <u>futhi</u> nezinye uqweqwe olunambitheka_ulamula...

Existential Quantification

- Different context: Option I in Table 1 for type (E0) Option II to axioms of type (E1)

(E0)	Ezulwini kune zingilosi	('in heaven there exist angels')
(E1)	Giraffe $\sqsubseteq$ $\exists$eats.Twig	
	yonke indlulamithi idla ihlamvana elilodwa	('each giraffe eats at least one twig')
	zonke izindlulamithi zidla ihlamvana elilodwa	('all giraffes eat at least one twig')
	yonke indlulamithi idla noma yiliphi ihlamvana	('each giraffe eats some twig')
	zonke izindlulamithi zidla noma yiliphi ihlamvana	('all giraffes eat some twig')
	yonke indlulamithi idla ihlamvanathize	('each giraffe eats some twig')

Beakdown–examples

noun	NC	RC	QC	QSuffix	copulative	EP	ESuffix
ihlamvana ('twig')	class 5	*eli-*	*-lo-*	*-dwa*			
isifundo ('module')	class 7	*esi-*	*-so-*	*-dwa*			
ushizi ('cheese')	class 3a	*o-*	*-ye-*	*-dwa*			
ihlamvana ('twig')	class 5				*yi-*	*-li-*	*-phi*
isifundo ('module')	class 7				*yi-*	*-si-*	*-phi*
ushizi ('cheese')	class 3a				*ngu-*	*-mu-*	*-phi*

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

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

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

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

Possible patterns for existential quantification

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.

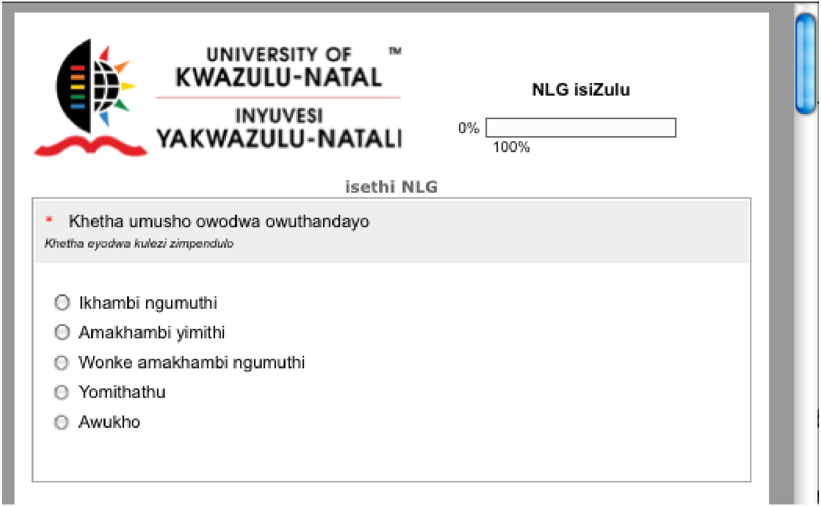
b. <All-concord for NC$_x$>onke <pl. N_1, is in NC$_x$> <conjugated verb> noma <copulative *ng/y* adjusted to first letter of N_2><EP of NC$_y$>phi <N_2>.

c. <All-concord for NC$_x$>onke <N_1 in NC$_x$> <conjugated verb> <N_2>thize;

Which options to choose?

- Survey, asking linguists and non-linguists for their preferences
- 10 questions pitting the patterns against each other
- Online, with isiZulu-localised version of Limesurvey (created as part of COMMUTERM project)
  - i.e., all text, buttons, autotext and error messages in isiZulu
- Analyse results in MS Excel

Question 1 (screenshot)



http://limesurvey.cs.ukzn.ac.za/index.php?sid=25965&lang=zu

Results

- 25 invited: students, academics (linguists), and non-linguists (such as administrators)
- 12 respondents: 5 linguists, 7 non-linguists (survey is still open)
- more agreement among linguists
- some differences possibly due to dialect
- preference for singular in subsumption
- other times plural
- other times also with universal quantification in the verbalization
- clear preference for the *-dwa* option

Results

- 25 invited: students, academics (linguists), and non-linguists (such as administrators)
- 12 respondents: 5 linguists, 7 non-linguists (survey is still open)
- more agreement among linguists
- some differences possibly due to dialect
- preference for singular in subsumption
- other times plural
- other times also with universal quantification in the verbalization
- clear preference for the *-dwa* option

Results

Question		Respondent			Question		Respondent		
		Ling.	Non-Ling.	Total			Ling.	Non-Ling.	Total
1. isa	sing.	80	0	33	6. exists	sing.+noma-phi	0	29	17
	pl.	0	43	25		pl.+noma-phi	0	0	0
	all+pl.	0	0	0		either	20	0	8
	either	20	57	42		neither	80	71	75
	neither	0	0	0					
2. isa	sing.	80	86	83	7. exists	sing.+-dwa	20	14	17
	pl.	0	0	0		pl.+-dwa	20	57	42
	all+pl.	0	0	0		either	40	0	17
	either	0	14	8		neither	20	29	25
	neither	20	0	8					
3. disj.	sing.	40	29	33	8. exists	sing.+-dwa	0	14	8
	all+pl.	0	14	8		sing.+noma-phi	20	0	8
	either	40	14	25		pl.+noma-phi	80	57	67
	neither	20	43	33		either	0	0	0
						neither	0	29	17
4. disj.	sing.	40	71	58	9. exists	pl.+noma-phi	40	14	25
	pl.	0	0	0		pl.+-thize	0	29	17
	either	20	0	8		either	40	43	42
	neither	40	29	33		neither	20	14	16
5. exists	pl.+-dwa	100	57	75	10. and	kanye	0	0	0
	pl.+noma-phi	0	14	8		futhi	0	14	8
	either	0	0	0		either	20	0	8
	neither	0	29	17		neither	80	86	83

Algorithm 1 Determine the verbalization of simple taxonomic subsumption

$\mathcal{C}$ set of classes, language $\mathcal{L}$ with $\sqsubseteq$ for subsumption and $\neg$ for negation; variables: A axiom, NC_i nounclass, $c_1, c_2 \in \mathcal{C}$, a_1 term, a_2 letter; functions: $getFirstClass(A)$, $getSecondClass(A)$, $getNC(C)$, $pluralizeNoun(C, NC_i)$, $checkNegation(A)$, $getFirstChar(C)$, $getNSC(NC_i)$, $getPNC(NC_i)$.

Require: axiom A with a $\sqsubseteq$ has been retrieved

$c_1 \leftarrow getFirstClass(A)$ {get subclass}
$c_2 \leftarrow getSecondClass(A)$ {get superclass}
$NC_1 \leftarrow getNC(c_1)$ {determine noun class by augment and prefix or dictionary}
$NC_2 \leftarrow getNC(c_2)$ {determine noun class by augment and prefix or dictionary}
if $checkNegation(A) = true$ **then**
  $NC'_1 \leftarrow$ lookup plural nounclass of NC_1 {from known list}
  $c'_1 \leftarrow pluralizeNoun(c_1, NC'_1)$
  $a_1 \leftarrow$ lookup quantitative concord for NC'_1 {from quantitative concord (QC(all)) list}
  $n \leftarrow getNSC(NC'_1)$ {get negative subject concord for c'_1}
  $p \leftarrow getPNC(NC_2)$ {get pronomial for c_2}
  RESULT $\leftarrow$ ‘ $a_1\ c'_1\ np\ c_2$. ’ {verbalise the disjointness}
else
  $a_2 \leftarrow getFirstChar(c_2)$ {retrieve first letter of c_2}
  select case
    $a_2 =$ ‘i’ **then**
      RESULT $\leftarrow$ ‘ c_1 yc_2 ’ {verbalise as taxonomic subsumption with y}
    $a_2 = \{$‘a’, ‘o’, ‘u’$\}$ **then**
      RESULT $\leftarrow$ ‘ c_1 ngc_2 ’ {verbalise as taxonomic subsumption with ng}
    $a_2 \notin \{$‘a’, ‘i’, ‘o’, ‘u’,$\}$ **then**
      RESULT $\leftarrow$ ‘this is not a well-formed isiZulu noun’
  end select case
end if
return RESULT

Algorithm 1 Determine the verbalization of simple taxonomic subsumption

$\mathcal{C}$ set of classes, language $\mathcal{L}$ with $\sqsubseteq$ for subsumption and $\neg$ for negation; variables: A axiom, NC_i nounclass, $c_1, c_2 \in \mathcal{C}$, a_1 term, a_2 letter; functions: $getFirstClass(A)$, $getSecondClass(A)$, $getNC(C)$, $pluralize$ … $tion(A)$, $getFirstChar(C)$, $getNSC(NC_i)$, $getPNC($ …

retrieve class and get its noun class

Require: axiom A with a $\sqsubseteq$ has been retrieved

$c_1 \leftarrow getFirstClass(A)$ {get subclass}

$c_2 \leftarrow getSecondClass(A)$ {get superclass}

$NC_1 \leftarrow getNC(c_1)$ {determine noun class by augment and prefix or dictionary}

$NC_2 \leftarrow getNC(c_2)$ {determine noun class by augment and prefix or dictionary}

if $checkNegation(A) = true$ **then**

  $NC'_1 \leftarrow$ lookup plural nounclass of NC_1 {from known list}

  $c'_1 \leftarrow pluralizeNoun(c_1, NC'_1)$

  $a_1 \leftarrow$ lookup quantitative concord for NC'_1 {from quantitative concord (QC(all)) list}

  $n \leftarrow getNSC(NC'_1)$ {get negative subject concord for c'_1}

  $p \leftarrow getPNC(NC_2)$ {get pronomial for c_2}

  RESULT $\leftarrow$ ' a_1 c'_1 np c_2. ' {verbalise the disjointness}

else

  $a_2 \leftarrow getFirstChar(c_2)$ {retrieve first letter of c_2}

  select case

    $a_2 =$ 'i' **then**

      RESULT $\leftarrow$ ' c_1 yc_2 ' {verbalise as taxonomic subsumption with y}

    $a_2 =$ {'a', 'o', 'u'} **then**

      RESULT $\leftarrow$ ' c_1 ngc_2 ' {verbalise as taxonomic subsumption with ng}

    $a_2 \notin$ {'a', 'i', 'o', 'u',} **then**

      RESULT $\leftarrow$ 'this is not a well-formed isiZulu noun'

  end select case

end if

return RESULT

Algorithm 1 Determine the verbalization of simple taxonomic subsumption

$\mathcal{C}$ set of classes, language $\mathcal{L}$ with $\sqsubseteq$ for subsumption and $\neg$ for negation; variables: A axiom, NC_i nounclass, $c_1, c_2 \in \mathcal{C}$, a_1 term, a_2 letter; functions: $getFirstClass(A)$, $getSecondClass(A)$, $getNC(C)$, $pluralizeNoun(C, NC_i)$, $checkNegation(A)$, $getFirstChar(C)$, $getNSC(NC_i)$, $getPNC(NC_i)$.

Require: axiom A with a $\sqsubseteq$ has been retrieved

```
c₁ ← getFirstClass(A)                                  {get subclass}
c₂ ← getSecondClass(A)                                 {get superclass}
NC₁ ← getNC(c₁)        {determine noun class by augment and prefix or dictionary}
NC₂ ← getNC(c₂)        {determine noun class by augment and prefix or dictionary}
if checkNegation(A) = true then
    NC'₁ ← lookup plural nounclass of NC₁              {from known list}
    c'₁ ← pluralizeNoun(c₁, NC'₁)
    a₁ ← lookup quantitative concord for NC'₁   {from quantitative concord (QC(all)) list}
    n ← getNSC(NC'₁)                                   {get ... rd for c'₁}
    p ← getPNC(NC₂)                                    {... ial for c₂}
    RESULT ← ' a₁ c'₁ np c₂. '                         {v... alise the disjointness}
else
    a₂ ← getFirstChar(c₂)                              {retrieve first letter of c₂}
    select case
        a₂ = 'i' then
            RESULT ← ' c₁ yc₂ '          {verbalise as taxonomic subsumption with y}
        a₂ = {'a', 'o', 'u'} then
            RESULT ← ' c₁ ngc₂ '         {verbalise as taxonomic subsumption with ng}
        a₂ ∉ {'a', 'i', 'o', 'u',} then
            RESULT ← 'this is not a well-formed isiZulu noun'
    end select case
end if
return RESULT
```

'simple' ISA

Algorithm 1 Determine the verbalization of simple taxonomic subsumption

$\mathcal{C}$ set of classes, language $\mathcal{L}$ with $\sqsubseteq$ for subsumption and $\neg$ for negation; variables: A axiom, NC_i nounclass, $c_1, c_2 \in \mathcal{C}$, a_1 term, a_2 letter; functions: $getFirstClass(A)$, $getSecondClass(A)$, $getNC(C)$, $pluralizeNoun(C, NC_i)$, $checkNegation(A)$, $getFirstChar(C)$, $getNSC(NC_i)$, $getPNC(NC_i)$.

Require: axiom A with a $\sqsubseteq$ has been retrieved

$c_1 \leftarrow getFirstClass(A)$ {get subclass}
$c_2 \leftarrow getSecondClass(A)$ {... erclass}
$NC_1 \leftarrow getNC(c_1)$ {determine ... ictionary}
$NC_2 \leftarrow getNC(c_2)$ {determine noun class b... gment and prefix or dictionary}
if $checkNegation(A) = true$ **then**
  $NC_1' \leftarrow$ lookup plural nounclass of NC_1 {from known list}
  $c_1' \leftarrow pluralizeNoun(c_1, NC_1')$
  $a_1 \leftarrow$ lookup quantitative concord for NC_1' {from quantitative concord (QC(all)) list}
  $n \leftarrow getNSC(NC_1')$ {get negative subject concord for c_1'}
  $p \leftarrow getPNC(NC_2)$ {get pronomial for c_2}
  RESULT $\leftarrow$ ‘ a_1 c_1' np c_2. ’ {verbalise the disjointness}
else
  $a_2 \leftarrow getFirstChar(c_2)$ {retrieve first letter of c_2}
  select case
    $a_2 =$ ‘i’ **then**
      RESULT $\leftarrow$ ‘ c_1 yc_2 ’ {verbalise as taxonomic subsumption with y}
    $a_2 =$ {‘a’, ‘o’, ‘u’} **then**
      RESULT $\leftarrow$ ‘ c_1 ngc_2 ’ {verbalise as taxonomic subsumption with ng}
    $a_2 \notin$ {‘a’, ‘i’, ‘o’, ‘u’,} **then**
      RESULT $\leftarrow$ ‘this is not a well-formed isiZulu noun’
  end select case
end if
return RESULT

negation (disjointness)

Algorithm 2 Determine the verbalization of conjunction in an axiom

$\mathcal{R}$ is the set of relationships, $\mathcal{A}$ of attributes, $\mathcal{C}$ of classes, and language $\mathcal{L}$ uses $\sqcap$ to denote conjunction; variables: e_2, c_1 a letter, A axiom; functions: $getNextVocabularyElement(A)$, $getFirstChar(e_2)$.

Require: axiom with a $\sqcap$ has been retrieved and position in string is known

```
e_2 ← getNextVocabularyElement(A)                {retrieve element after the ⊓}
if e_2 ∈ R ∪ A then
    RESULT ← ' kanye '                           {verbalise ⊓ as kanye}
else
    if e_2 ∈ C then
        c_1 ← getFirstChar(e_2)                  {retrieve first letter of e_2}
        select case
            c_1 = 'i' then
                e_2^- ← drop c_1 from e_2
                RESULT ← ' nee_2^- '             {verbalise ⊓ with ne- prefix}
            c_1 = 'u' then
                e_2^- ← drop c_1 from e_2
                RESULT ← ' noe_2^- '             {verbalise ⊓ with no- prefix}
            c_1 = 'a' then
                e_2^- ← drop c_1 from e_2
                RESULT ← ' nae_2^- '             {verbalise ⊓ with na- prefix}
            c_1 ∉ {'i', 'u', 'a'} then
                RESULT ← 'this is not a well-formed isiZulu noun'
        end select case
    else
        RESULT ← 'this is not a well-formed axiom'
    end if
end if
return RESULT
```

Algorithm 2 Determine the verbalization of conjunction in an axiom

$\mathcal{R}$ is the set of relationships, $\mathcal{A}$ of attributes ... language $\mathcal{L}$ uses $\sqcap$ to denote conjunction; variables: ... tions: $getNextVocabularyElement(A)$, $getFirstCh$...

enum-and or conn-and?

Require: axiom with a $\sqcap$ has been retrieved and positio... string is known

$e_2 \leftarrow getNextVocabularyElement(A)$ {retrieve element after the $\sqcap$}
if $e_2 \in \mathcal{R} \cup \mathcal{A}$ **then**
  RESULT ← ' kanye ' {verbalise $\sqcap$ as kanye}
else
  if $e_2 \in \mathcal{C}$ **then**
    $c_1 \leftarrow getFirstChar(e_2)$ {retrieve first letter of e_2}
    select case
      c_1 = 'i' **then**
        $e_2^- \leftarrow$ drop c_1 from e_2
        RESULT ← ' nee_2^- ' {verbalise $\sqcap$ with ne- prefix}
      c_1 = 'u' **then**
        $e_2^- \leftarrow$ drop c_1 from e_2
        RESULT ← ' noe_2^- ' {verbalise $\sqcap$ with no- prefix}
      c_1 = 'a' **then**
        $e_2^- \leftarrow$ drop c_1 from e_2
        RESULT ← ' nae_2^- ' {verbalise $\sqcap$ with na- prefix}
      $c_1 \notin$ {'i', 'u', 'a'} **then**
        RESULT ← 'this is not a well-formed isiZulu noun'
    end select case
  else
    RESULT ← 'this is not a well-formed axiom'
  end if
end if
return RESULT

Algorithm 2 Determine the verbalization of conjunction in an axiom

$\mathcal{R}$ is the set of relationships, $\mathcal{A}$ of attributes, $\mathcal{C}$ of classes, and language $\mathcal{L}$ uses $\sqcap$ to denote conjunction; variables: e_2, c_1 a letter, A axiom; functions: $getNextVocabularyElement(A)$, $getFirstChar(e_2)$

Require: axiom with a $\sqcap$ has been retrieved and positi…

connective-and

$e_2 \leftarrow getNextVocabularyElement(A)$ {…ieve element after the $\sqcap$}

if $e_2 \in \mathcal{R} \cup \mathcal{A}$ **then**

  RESULT ← ' kanye ' {verbalise $\sqcap$ as kanye}

else

  if $e_2 \in \mathcal{C}$ **then**

    $c_1 \leftarrow getFirstChar(e_2)$ {retrieve first letter of e_2}

    select case

      c_1 = 'i' **then**

        $e_2^- \leftarrow$ drop c_1 from e_2

        RESULT ← ' nee_2^- ' {verbalise $\sqcap$ with ne- prefix}

      c_1 = 'u' **then**

        $e_2^- \leftarrow$ drop c_1 from e_2

        RESULT ← ' noe_2^- ' {verbalise $\sqcap$ with no- prefix}

      c_1 = 'a' **then**

        $e_2^- \leftarrow$ drop c_1 from e_2

        RESULT ← ' nae_2^- ' {verbalise $\sqcap$ with na- prefix}

      $c_1 \notin$ {'i', 'u', 'a'} **then**

        RESULT ← 'this is not a well-formed isiZulu noun'

    end select case

  else

    RESULT ← 'this is not a well-formed axiom'

  end if

end if

return RESULT

Algorithm 2 Determine the verbalization of conjunction in an axiom

$\mathcal{R}$ is the set of relationships, $\mathcal{A}$ of attributes, $\mathcal{C}$ of classes, and language $\mathcal{L}$ uses $\sqcap$ to denote conjunction; variables: e_2, c_1 a letter, A axiom; functions: $getNextVocabularyElement(A)$, $getFirstChar(e_2)$.

Require: axiom with a $\sqcap$ has been retrieved and position in string is known

```
e_2 ← getNextVocabularyElement(A)                    {retrieve element after the ⊓}
if e_2 ∈ R ∪ A then
    RESULT ← ' kanye '                                {…s kanye}
else
    if e_2 ∈ C then
        c_1 ← getFirstChar(e_2)                       {retrieve first letter of e_2}
        select case
            c_1 = 'i' then
                e_2^- ← drop c_1 from e_2
                RESULT ← ' nee_2^- '                  {verbalise ⊓ with ne- prefix}
            c_1 = 'u' then
                e_2^- ← drop c_1 from e_2
                RESULT ← ' noe_2^- '                  {verbalise ⊓ with no- prefix}
            c_1 = 'a' then
                e_2^- ← drop c_1 from e_2
                RESULT ← ' nae_2^- '                  {verbalise ⊓ with na- prefix}
            c_1 ∉ {'i', 'u', 'a'} then
                RESULT ← 'this is not a well-formed isiZulu noun'
        end select case
    else
        RESULT ← 'this is not a well-formed axiom'
    end if
end if
return RESULT
```

enumerative-and

Algorithm 3 Determine the verbalization of existential quantification with object property (first, basic, version)

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

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

$c_1 \leftarrow getFirstClass(A)$ {get subclass}
$c_2 \leftarrow getSecondClass(A)$ {get superclass}
$o \leftarrow getObjProp(A)$ {get object property}
$NC_1 \leftarrow getNC(c_1)$ {determine noun class by augment and prefix or dictionary}
$NC_2 \leftarrow getNC(c_2)$ {determine noun class by augment and prefix or dictionary}
$NC_1' \leftarrow$ lookup plural nounclass of NC_1 {from known list}
$c_1' \leftarrow pluralizeNoun(c_1, NC_1')$
$a_1 \leftarrow$ lookup quantitative concord for NC_1' {from quantitative concord (QC(all)) list}
$o' \leftarrow AlgoConjugate(o, NC_1)$ {call algorithm *AlgoConjugate* to conjugate o}
$r_2 \leftarrow getRC(NC_2)$ {get relative concord for c_2}
$q_2 \leftarrow getQC(NC_2)$ {get quantitative concord for c_2 from the QC_{dwa}-list}
RESULT $\leftarrow$ ' a_1 c_1' o' c_2 $r_2 q_2$dwa. ' {verbalise the simple axiom}
return RESULT

Algorithm 3 Determine the verbalization of existential quantification with object property (first, basic, version)

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

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

$c_1 \leftarrow getFirstClass(A)$ {get subclass}
$c_2 \leftarrow getSecondClass(A)$ {get superclass}
$o \leftarrow getObjProp(A)$ {get object property}
$NC_1 \leftarrow getNC(c_1)$ {determine noun class by augment and prefix or dictionary}
$NC_2 \leftarrow getNC(c_2)$ {determine noun class by augment and prefix or dictionary}
$NC_1' \leftarrow$ lookup plural nounclass of NC_1 {...n list}
$c_1' \leftarrow pluralizeNoun(c_1, NC_1')$
~~$a_1 \leftarrow$ lookup quantitative concord for NC_1' {from quantitative concord (QC(all)) list}~~
$o' \leftarrow AlgoConjugate(o, NC_1)$ {call algorithm $AlgoConjugate$ to conjugate o}
~~$r_2 \leftarrow getRC(NC_2)$~~ {get relative concord for c_2}
$q_2 \leftarrow getQC(NC_2)$ {get quantitative concord for c_2 from the QC_{dwa}-list}
RESULT $\leftarrow$ ' a_1 c_1' o' c_2 $r_2q_2$dwa. ' {verbalise the simple axiom}
return RESULT

to be done...

Example

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

Example

- $\forall x$ (Professor(x) $\rightarrow \exists y$ (teaches(x, y) $\wedge$ Course(y)))
- Professor $\sqsubseteq \exists$ teaches.Course
- **Each** Professor teaches **at least one** Course
- $\forall x$ (uSolwazi(x) $\rightarrow \exists y$ (ufundisa(x, y) $\wedge$ Isifundo(y)))
- uSolwazi $\sqsubseteq \exists$ ufundisa.Isifundo
- ?

$$\forall x\ (\mathrm{uSolwazi}(x) \rightarrow \exists y\ (\mathrm{ufundisa}(x, y) \land \mathrm{Isifundo}(y)))$$

$$\mathrm{uSolwazi} \sqsubseteq \exists\ \mathrm{ufundisa}.\mathrm{Isifundo}$$

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

uSolwazi $\sqsubseteq \exists$ ufun…

look-up NC

pluralise

for-all

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

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

Bonke oSolwazi

$$\forall x\ (uSolwazi(x) \rightarrow \exists y\ (ufundisa(x, y) \land Isifundo(y)))$$

$uSolwazi \sqsubseteq \exists ufundisa.$

... for relevant NC. Here:

ngi-

u-

u-

si-

ni-

ba-

AlgoConjugate

Bonke oSolwazi bafundisa

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

$\text{uSolwazi} \sqsubseteq \exists\ \text{ufundisa}.\text{Isifundo}$

Bonke oSolwazi bafundisa Isifundo

$\forall x\,(uSolwazi(x) \rightarrow \exists y\,(\ \ldots\ ) \land \ \ldots\ ))$

$uSolwazi \sqsubseteq \exists$ ufundisa.I…

look-up NC

get RC

get QC

add -dwa

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

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

Bonke oSolwazi bafundisa Isifundo esisodwa

Outline

Discussion

- Template-based approach is not applicable to isiZulu (and, more generally: Bantu languages that have noun classes)
  - Or: grammar engine needed
- Devising the patterns hampered by outdated literature
- Several preferences for patterns
- Algorithms nontrivial; covering:
  - 'simple' existential and universal quantification
  - taxonomic subsumption
  - negation (class disjointness)
  - conjunction

Discussion

- Template-based approach is not applicable to isiZulu (and, more generally: Bantu languages that have noun classes)
    - Or: grammar engine needed
- Devising the patterns hampered by outdated literature
- Several preferences for patterns
- Algorithms nontrivial; covering:
    - 'simple' existential and universal quantification
    - taxonomic subsumption
    - negation (class disjointness)
    - conjunction

Some other potential use: machine translation

- Google Translate English-isiZulu translates, e.g., "mix the sugar and milk and butter" as "*hlanganisa ushukela nobisi ibhotela*" (translation d.d. 14-1-2014)
  - Misses the second conjunction in the enumeration
  - ushukela ⊓ ubisi ⊓ ibhotela with Algorithm for conjunction obtains correct verbalisation/translation: *ushukela nobisi nebhotela*
- Google's "all giraffes eat twigs" is translated as "*yonke izindlulamithi udle amahlumela*" (translation d.d. 14-1-2014)
  - But *izindlulamithi* is in noun class 10, not 9, so it goes with *zonke*
  - This can be correctly verbalised following Algorithm subsumption verbalization (line 9).

Outline

Conclusions

- Novel verbalization patterns and algorithms for simple subsumption, disjoint classes, conjunction, and basic options with quantification
- Verbalizing formally represented knowledge in isiZulu requires a grammar engine even for the relatively basic language constructs
- Due to, principally: i) the system of noun classes, ii) the system of complex agreement, iii) phonological conditioned copulatives, and iv) verb conjugation
- The survey on verbalization pattern preference showed a clear preference for the *-dwa* option, and more variation in preference by the non-linguists

Future work

- To be done for ‘full’ OWL 2 EL and $\mathcal{ALC}$, mainly:
    - Transitivity
    - More elaborate axioms, such as $\forall R.C \sqsubseteq \exists S.(D \sqcap E)$
    - Negation in other cases
    - Union
- Conjugation of verbs present and past tense, and the prepositions (*taught* by, works *for*)
- Preference of patterns vs understandability
- Living vs. non-living thing distinction
- Interaction with multilingual ontologies (e.g., extending *Lemon* [McCrae et al.(2012)])
- Implement it

References I

Ronell Alberts, Thomas Fogwill, and C. Maria Keet.
Several required OWL features for indigenous knowledge management systems.
In P. Klinov and M. Horridge, editors, *7th Workshop on OWL: Experiences and Directions (OWLED 2012)*, volume 849 of *CEUR-WS*, page 12p, 2012.
27-28 May, Heraklion, Crete, Greece.

F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors.
The Description Logics Handbook – Theory and Applications.
Cambridge University Press, 2 edition, 2008.

M. Curland and T. Halpin.
Model driven development with NORMA.
In *Proceedings of the 40th International Conference on System Sciences (HICSS-40)*, pages 286a–286a. IEEE Computer Society, 2007.
Los Alamitos, Hawaii.

Enrico Franconi, Paolo Guagliardo, and Marco Trevisan.
An intelligent query interface based on ontology navigation.
In *Workshop on Visual Interfaces to the Social and Semantic Web (VISSW'10)*, 2010.
Hong Kong, February 2010.

Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn.
Discourse Representation Structures for ACE 6.6.
Technical Report ifi-2010.0010, Dept of Informatics, University of Zurich, Switzerland, 2010.

Mustafa Jarrar, C. Maria Keet, and Paolo Dongilli.
Multilingual verbalization of ORM conceptual models and axiomatized ontologies.
Starlab technical report, Vrije Universiteit Brussel, Belgium, February 2006.

References II

C.M. Keet and L. Khumalo.
Toward verbalizing logical theories in isiZulu.
In *Proceedings of the 4th Workshop on Controlled Natural Language (CNL'14)*, LNAI, page (accepted). Springer, 2014a.
20-22 August 2014, Galway, Ireland.

C.M. Keet and L. Khumalo.
Basics for a grammar engine to verbalize logical theories in isiZulu.
In *Proceedings of the 8th International Web Rule Symposium (RuleML'14)*, LNCS, page (accepted). Springer, 2014b.
August 18-20, 2014, Prague, Czech Republic.

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

John McCrae, Guadalupe Aguado de Cea, Paul Buitelaar, Philipp Cimiano, Thierry Declerck, Asunción Gómez-Pérez, Jorge Gracia, Laura Hollink, Elena Montiel-Ponsoda, Dennis Spohr, and Tobias Wunner.
The lemon cookbook.
Technical report, Monnet Project, June 2012.
www.lemon-model.net.

R. Schwitter, K. Kaljurand, A. Cregan, C. Dolbear, and G. Hart.
A comparison of three controlled natural languages for OWL 1.1.
In *Proceedings of OWL: Experiences and Directions (OWLED'08 DC)*, 2008.
Washington, DC, USA metropolitan area, on 1-2 April 2008.

References III

Allan Third, Sandra Williams, and Richard Power.
OWL to English: a tool for generating organised easily-navigated hypertexts from ontologies.
poster/demo paper, 2011.
10th International Semantic Web Conference (ISWC'11), 23-27 Oct 2011, Bonn, Germany.

Thank you!