Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2846–2854
Marseille, 11–16 May 2020

Towards Computational Resource Grammars for Runyankore and Rukiga

David Bamutura, Peter Ljunglöf, Peter Nabende
Chalmers Univ. of Tech. & Mbarara Univ. of Sci. & Tech. , Univ. of Gothenburg , Makerere University
bamutra@chalmers.se, peter.ljunglof@cse.gu.se, peter.nabende@gmail.com

Abstract

In this paper, we present computational resource grammars of Runyankore and Rukiga (R&R) languages. Runyankore and Rukiga are two under-resourced Bantu Languages spoken by about 6 million people indigenous to South Western Uganda, East Africa. We used Grammatical Framework (GF), a multilingual grammar formalism and a special-purpose functional programming language to formalise the descriptive grammar of these languages. To the best of our knowledge, these computational resource grammars are the first attempt to the creation of language resources for R&R. In Future Work, we plan to use these grammars to bootstrap the generation of other linguistic resources such as multilingual corpora that make use of data-driven approaches to natural language processing feasible. In the meantime, they can be used to build Computer-Assisted Language Learning (CALL) applications for these languages among others.

Keywords: Grammar, Syntax, Morphology, Runyankore, Rukiga, less-resourced Languages, Grammatical Framework, Resource Grammar Library

1. Introduction

Runyankore & Rukiga (hereafter R&R) are two heavily under-resourced Bantu languages. Their limited presence on the web makes it difficult to develop substantial computational linguistic resources for these languages. Consequently, the lack of such resources makes the use of data-driven Natural Language Processing (NLP) approaches unsuitable for these languages. However, rule-based approaches such as grammars, can be used to bootstrap the creation of such resources. In this paper we present computational resource grammars of these two languages developed using Grammatical Framework (GF).

1.1. Grammatical Framework (GF)

GF is a multilingual grammar formalism, a logical framework and a special-purpose functional programming language for defining grammars of both formal and natural languages (Ranta, 2011; Ranta, 2009a). We chose GF because it does not need any additional linguistic resources, and being multilingual, it can be used to develop resources for under-resourced languages by using existing linguistic resources of well-resourced languages already covered in its Resource Grammar Library (RGL) (Ranta, 2009a; Kolachina and Ranta, 2016).

1.2. Abstract and Concrete Syntax

Each grammar in GF consists of an *abstract and concrete syntax*. The abstract syntax defines a set of abstract syntactic structures, called abstract terms or trees, which are used to define a language-independent or semantic meaning representation. The concrete syntax defines a relation between the abstract structures and their language-specific constructions. This makes it possible to define several sets of concrete "syntaxes" for one single abstract syntax. The single abstract syntax then acts as an interlingua between different languages. The concept of a shared abstract syntax is the reason for the multilingual capabilities of GF.

1.3. Resource & Application Grammars

Grammars designed in GF are of two types: *resource* and *application grammars*. Resource grammars are broad-coverage grammars developed from scratch for the purpose of formally describing the morphology and syntax of natural languages while application grammars model semantic information about a specific application domain. Using GF's modular system, Resource Grammars are packaged together and exposed by both a common API (that is based on the common abstract syntax) and language specific APIs into what is called the GF Resource Grammar Library (GF-RGL) (Ranta, 2009b). Application grammars make use of general linguistic functions implemented in resource grammars by accessing them through the GF-RGL. Resource grammars have been used successfully in domain-limited application areas such as Multilingual Document Authoring (Dymetman et al., 2000), low-coverage multilingual translation (Ranta et al., 2010), domain specific dialogue systems such as music players (Perera and Ranta, 2007) and Computer-Assisted Language Learning (CALL) (Lange, 2018; Lange and Ljunglöf, 2018). Another important use case in the area of localisation is the multilingual dissemination of weather information especially in multilingual societies. Our immediate motivation is therefore to utilise the GF-RGL for R&R to leverage the work done by Lange (2018) on CALL for the Latin language in order to build, localise and improve tools that can be used to create automatic exercises for learning R&R grammar to higher levels of proficiency accessible to all.

In the rest of the paper, we discuss related work in Section 2., an overview of R&R, its nominal and verbal morphology in Section 3. followed the GF-RGL implementation of R&R in section 4.. Sections 5.,6. and 7. present Observations through an example, Discussion and lastly Conclusion & Future Work respectively.

2. Related Work

Previous work on the computational modelling of the grammar of R&R include: noun and verb morphological analysers by Katushemererwe and Hanneforth (2010b; 2010a), a Controlled Natural Language for Runyankore (Byamugisha et al., 2016) and a Noun pluralizer (Byamugisha et al., 2018). However, this work has been limited to small fragments of the languages. Within the GF community, there has been work on computational modeling of Bantu languages: Kikamba (Kituku et al., 2019), Tswana (Pretorius et al., 2017), and Swahili (Ngángá, 2012). While we consulted the Swahili implementation during initial development, we found that Swahili is morphologically and syntactically less complex than R&R. Additionally, its coverage of the GF-RGL functions was very small. Little insight was generated from that grammar. Likewise the Tswana GF-RGL was limited to modelling the proper verb for declarative sentences which is small in scope. Twsana's use of both a disjunctive and conjunctive orthography as compared to R&R's conjunctive morphology also provided limited insights into how to implement the grammars of R&R. Work on Kikamba and R&R was done during the same timeframe and hence both of us benefited from the sharing of ideas.

3. Runyankore & Rukiga (R&R)

R&R are languages spoken in South-Western Uganda by about 6 million people (Simons and Fennig, 2018). They belong to the **JE10** zone (Maho, 2009) of the Niger-Congo Bantu language family. Just like any other Bantu languages, morphologically, R&R are **highly agglutinating** (e.g., the single word *tinkamureebagaho* (ti-n-ka-mu-reeb-a-ga-ho) is a sentence meaning "I have never seen him/her"), exhibit high instances of **phonological conditioning** and a **large Noun Class System** of 17 noun classes (Katushemererwe and Hanneforth, 2010b; Byamugisha et al., 2016). This noun class system dictates a complex concordial system of agreement among phrasal categories. These properties make the morphology of the languages more complex to computationally model as compared to analytic languages such as English. Since both languages share the same dictionaries (Taylor and Yusuf, 2009; Mpairwe and Kahangi, 2013a) and grammar books (Morris and Kirwan, 1972; Mpairwe and Kahangi, 2013b) their grammar is largely identical while the lexicon differs by 6%–16% (Turyamwomwe, 2011; Simons and Fennig, 2018).

3.1. Nominal Morphology

The morphological structure of nouns in R&R consists of two parts, a **class prefix** and a **noun stem**. The class prefix is further divided into an **Initial Vowel (IV)** and a **Noun Class particle (NCP)** (Mpairwe and Kahangi, 2013b). The **noun stem** usually bears the bulk of the semantic meaning of the noun. Each Noun in R&R, belongs to a particular **Noun Class (NC)**. The group of possible noun classes is given in Table 1 adapted from Katushemererwe and Hanneforth (2010b) with modifications. The predominant naming scheme of noun classes in Bantu languages (called the Bleek-Meinhoff system) makes use of a combination of a numeral and optionally letters (see column labelled Numbers in Table 1). However, we discovered an alternative scheme that uses NCP (refer to "Particles" column in the same table) utilised by (Mpairwe and Kahangi, 2013a; Mpairwe and Kahangi, 2013b) in their dictionary and grammar books. Since we make heavy use of these books, we have found it convenient to use the latter scheme in order to avoid an additional step of mapping between the two systems during our implementation of the grammar as explained in section 4. Apart from locative particles -ha-, -mu- and -ku- , most of the other particles can be arranged in singular-plural pairs for common nouns. We generalise such a pairing using the notation $[\alpha - \beta]$ where α & β are noun class particles chosen from the sets of singular & plural particles respectively. We borrow the use of the number ZERO (0) from Mpairwe and Kahangi (2013a) in their Runyankore-Rukiga dictionary to denote absence of either singularity or plurality in order to maintain the pairing for such nouns. Hence the pairs $[\alpha - 0]$, $[0 - \beta]$ and $[0 - 0]$ which represent nouns that are always singular, plural and those that collectively neither have an IV nor noun class particle respectively. It is important to note that classes 9_10 and 9 in the table are both assigned N_N because the set of agreement concords for the two classes are the same. More noun classes are used in our implementation to cater for Numerals which are a special set of nouns for naming entities used to count (**ordinals**), or encode order (**Ordinals**).

3.2. Verbal Morphology

In Meeussen's (1967) original construction, the Bantu verbal unit consists of a **pre-stem** and **stem**. The stem is further divided into a **base** and **final vowel (FV)**. The base is also divided into a **radical (Rad)** and **extensions**. Further subdivisions in each of these parts results into 11 slots (Katushemererwe and Hanneforth, 2010a; Turyamwomwe, 2011), with each slot taking a set of morphemes for a particular purpose such as Primary/Secondary negative (Pneg / Sneg), subject (**S**), object, tense, aspect and other markers. Regular verbs can be classified into four base-forms: Imperatives, Subjunctives, Perfectives and Infinitives. They can be rendered in active or passive voice and within each voice, the verb can take the form of Simple, Prepositional and Causative.

In the verbal unit of R&R, Tense and Aspect (T/A) are marked using morphemes which may be simple or compound. However, in our attempt to model the grammar of R&R, we have combined the constructions suggested by Muzale (1998), Katushemererwe and Hanneforth (2010a) and Turyamwomwe (2011), based on omissions and coverage made by each. While Muzale (1998) shows how different T/A markers have developed through time (diachronicaly) up to their current forms (as of 1998) among Rutara, Katushemererwe and Hanneforth (2010a) confine their work to Runyakitara and Turyamwomwe (2011) restricts himself to T/A in R&R. Therefore our design was based first on Muzale (1998) followed by Katushemererwe and Hanneforth (2010a) and lastly Turyamwomwe (2011) for verbs. Traditionally, tense is divided into Past and Present and Future. However, in R&R the past is split into the Remote Past, Near Past and Immediate Past (Turyamwomwe,

Class			Individual Particles		Example		Gloss
ID	Numbers	Particles	Singular	Plural	Singular	Plural	Singular(Plural)
1	1_2	MU_BA	MU	BA	o-mu-shaija	a-ba-shaija	man (men)
2	1a	MU_ZERO	MU	n/a	o-mu-hangi	n/a	creator (n/a)
3	1b/2b	ZERO_BAA	n/a	BAA	swhento	baa-shwento	Uncle(s)
4	3_4	MU_MI	MU	MI	o-mu-ti	e-mi-ti	tree(s)
5	3a	MU_ZERO	MU	n/a	o-mwisyo	n/a	breath (n/a)
6	4a	ZERO_MI	n/a	MI	n/a	e-mi-gyendere	n/a (way of walking)
7	5_6	RI_MA	RI	MA	e-ri-sho	a-ma-isho	eye(s)
8	5a	I_MA	I	MA	e-i-teeka	a-ma-teeka	law(s)
9	5b	I_ZERO	I	n/a	e-i-tétsi	n/a	pampering(n/a)
10	6a	ZERO_MA	n/a	MA	n/a	a-ma-te	milk (milk)
11	7_8	KI_BI	KI	BI	e-ki-ti	e-bi-ti	stick (stick)
12	7	KI_ZERO	KI	n/a	e-ki-niga	n/a	anger (n/a)
13	8	ZERO_BI	n/a	BI	n/a	e-bi-bembe	(n/a) leprosy
14	9_10	N_N	N	N	e-n-te	e-n-te	cow(s)
15	9	N_N	n/a	n/a	e-bahaasa	e-bahaasa	envelope(s)
16	10	ZERO_ZERO	n/a	n/a	bwîno	bwîno	ink (ink)
17	11_10	RU_N	RU	N	O-ru-shózi	e-n-shózi	mountain(s)
18	12_14	KA_BU	KA	BU	a-ká-bunza	o-bu-bunza	question mark(s)
19	12	KA_ZERO	KA	n/a	a-ka-bi	n/a	danger (n/a)
20	14	ZERO_BU	n/a	BU	n/a	o-bu-cécezi	n/a(being humble)
21	13	ZERO_TU	n/a	TU	n/a	o-tu-ro	n/a (sleep)
22	15_6	KU_MA	KU	MA	o-ku-guru	a-ma-guru	leg(s)
23	16	HA_ZERO	HA	n/a	a-ha-kaanyima(*)	n/a	behind the house (n/a)
24	17	KU_ZERO	KU	n/a	o-ku-z/'imu	n/a	Underground (n/a)
25	18	MU_ZERO	MU	n/a	o-mu-nda	n/a	in the stomach (n/a)
26	20_21	GU_GA	GU	GA	o-gu-kazi	a-ga-kazi	bad woman (women)
27	11_14	RU_BU	RU	BU	o-rur-o	o-bu-ro	one millet grain (many)
28	14_6	BU_MA	BU	MA	o-bu-ta	a-ma-ta	bow(s)
29	γ	RU_ZERO	RU	n/a	0-ru-me	n/a	dew (n/a)

Table 1: The Runyankore and Rukiga noun class system (both the numerical system and that based on Individual particles) and examples of both singular and plural. Adapted from Katushemererwe & Hanneforth (2010a) & updated using the dictionary by Mpairwe & Kahangi (2013)

2011). We found that the Memorial Present identified in Muzale (1998) and Immediate Past (Katushemererwe and Hanneforth, 2010a; Turyamwomwe, 2011) are one and the same i.e. they mean the same and use identical tense and polarity agreement markers. The tense markers for all these tenses are summarised in Table 2.

The Universal Tense is identical to Muzale's (1998) Experiential Present. The Future is divided into the Near and Far / Remote Future. As an example, Table 2 shows how different morphemes are combined to form a verb for the seven tenses while omitting markers for direct and indirect objects. With regard to Aspect, Muzale (1998) identifies Retrospective, Resultative, Persistive and Remote Retrospective in addition to Perfective, Progressive, Persistive and Habitual identifed by Turyamwomwe (2011).

3.3. Reason for lack of resources

Despite the initial exposure to learning R&R in the first three years of primary school, English becomes the official language of instruction and examination from the fourth year on, severely limiting the continued study of R&R to higher levels of proficiency. It is also worthy to note that although dictionaries, grammar books and an orthography for R&R exist, R&R just like any other native languages in Uganda largely remain oral as opposed to written even among those literate in English. Only a dismal few study the language to a level sufficient to achieve proficiency in writing which implies lack of continuity in learning the grammar of the language. This explains the nearly zero presence on the web hence the lack of any computational language resources. As a result, the languages are highly under-resourced. It is therefore important to take steps in building language resources, encouraging writing in these languages and their preservation.

4. GF-RGL Implementation of R&R

In this section, we explain how the grammars for R&R were implemented using GF. The GF-RGL does not attempt to cover all grammatical and morphological structures in all languages, but instead focus is put on constructions that are common amongst the many languages of the world. It implements more than 50 grammatical categories and almost

Universal Tense	Tense in R&R	Pol	"To see"	Generalization
Past	Remote Past	Pos	S-ka-reeb-a	S-ka-Rad-FV
		Neg	ti-S-rá-reeba -ir-e	Pneg-S-TM-Rad-TM-FV
	Near Past	Pos	S-∅-reeb-ir-e	S-∅-Rad-TM-FV
		Neg	ti-S-∅-reeb-ir-e	Pneg-S-∅-Rad-TM-FV
	Immediate Past	Pos	S-áá-reeb-a	S-TM-Rad-TM-e
		Neg	ti-S-áá-reeb-a	Pneg-S-TM-Rad-TM-FV
Present	Memorial Present	Pos	S-áá-reeb-a	S-TM-Rad-FV
		Neg	ti-S-áá-reeb-a	Pneg-S-TM-Rad-FV
	Experiential Present	Pos	S-∅-reeb-a	S-∅-Rad-FV
		Neg	ti-S-∅-reeb-a	Pneg-S-∅-Rad-Fv
Future	Near Future	Pos	ni-S-ija/za ku-reeb-a	CM-S-ija /za ku-Rad-FV
		Neg	ti-S-ku-ija/ku-za ku-reeb-a **or** ti-tu-ra-reeb-FV	Pneg-S-ku-ija /za ku-Rad-FV **or** Pneg-tu-ra-Rad-FV
	Remote Future	Pos	S-riá-reeba-a	S-TM-Rad-FV
		Neg	ti-S-riá-reeba-a	Pneg-S-TM-Rad-FV

Table 2: How different morphemes are combined to form a verb. CM = Continuous Tense Marker, Pneg = Primary Negative marker, Sneg= Secondary Negative marker, S = Subject Marker, followed by a Tense Marker (TM), ∅ = absence of TM, Rad = Radical and FV = Final Vowel. Note: Pos = Positive and Neg = Negative. The Immediate Past and memorial present are one and the same referring to an event the occurred a moment earlier.

200 construction functions. Because of the expressive module system of GF, it is possible to extend the common GF-RGL with language-specific constructions. The task is to write concrete modules for each abstract module.

4.1. Lexicon

When building an RGL for any language, the first thing to tackle is the lexicon. For each lexical item defined in the abstract module of the GF-RGL lexicon, a concrete mapping must be implemented for the language under investigation. This concrete mapping involves the enumeration of all possible morphological inflectional forms of the lemma provided. It is impossible to have a strict one-to-one mapping due to the existence of synonyms and lexical gaps. Synonyms are treated as separate GF lexical categories, so we selected a single word from the set of synonyms and left other synonyms to be catered for by an Extension module for the Lexicon. For lexical gaps in R&R which are a result of cultural differences, modernisation and lack of universality in language, we employed loan words (influenced by English) and adapted them according to the orthography of R&R. For the problem of a lack of a rich notion of adjectives particularly with respect to **degree**, we used circumscription. Just like GF-RGLs for other languages, we minimised the requirement of explicitly enumerating all the inflectional forms of a lexical item from a given category through the use of morphological paradigms. If a lexical entry ω of a given lexical type C has surface forms $\langle \omega_1, \omega_2 \ldots \omega_n \rangle$, then these paradigms are special functions that take between one surface form (base form) and at most $n-1$ surface forms and other information to produce the full set of inflected word-forms of that lexical entry. Paradigms that take one surface form, called smart paradigms (Détrez and Ranta, 2012), are restricted to lexemes whose inflection is regular.

4.1.1. Common Nouns and Proper Nouns

In R&R, common nouns inherently belong to a noun class. It is possible to use these nouns in either their **Complete** or **Incomplete** forms and each of these is inflected for number (refer to Table 3 for an example). We therefore declared parameters for NounState, Number and Gender (lines 2-4), a linearisation type for Nouns (lines 21-22) in code listing 1 on the next page. We also declared paradigms for computing inflection tables for nouns. We used a composite parametric data type similar to algebraic data types from functional programming to encode agreement with respect to noun class, Person and Number in lines 4-12 of listing 1. Under normal circumstances Proper Nouns do not inflect with number. They are all in the third person but belong to different noun classes based on the common noun they give a name to. It was therefore necessary to keep track of information about Agreement and whether the noun refers to a location or place (refer to lines 23-25 in lstlisting 1). The smart-paradigm we implemented for nouns (smartNoun) is a very accurate "pluraliser" which handles most of the cases using pattern-matching. Incomplete nouns are used to compose noun phrases from determiners and nouns for example ("every person" is realised as "buri muntu" with the initial vowel of "person" removed).

```
1  param
2      Number = Sg | Pl;
3      NounState  = Complete | Incomplete  ;
4      Gender = MU_BA | MU_ZERO ... RU_ZERO;
5      Case = Acc |  Nom |Gen;
6      ConjArg = Nn_Nn | Nps_Nps | Pns_Pns  |  RelSubjCls
7          | Other;
8      AgrConj = AConj ConjArg;
9      Agreement =  AgP3 Number Gender
10         | AgMUBAP1 Number | AgMUBAP2 Number
11         | NONE;
12     AgrExist  = AgrNo | AgrYes Agreement;
13     Position  = Post  | Pre;
14     RCase = RSubj |  RObj;
15   RForm = RF RCase | Such_That;
16   -- Possible Complement types held by a ClSlash
17   ComplType = Nn |Ap | Adverbial  |AdverbialVerb
18     | Empty;
19   VVFForm = VVImp | VVPerf | VVBoth;
20 oper
21   Noun : Type = {s :  Number⇒ NounState⇒ Str  ;
22     gender : Gender} ;
23   ProperNoun : Type =
24     {s: Str  ; a:Agreement ; isPlace  : Bool};
25   mkXClitic : Agreement → Str  = \a →
26       case a of {
27         AgMUBAP1 n ⇒mkClitics "n" "tu" n;
28         -- about 20−30 more table rows
29         ...
30       };
31   mkXCliticTable : Agreement ⇒ Str =
32     table {
33       AgMUBAP1 n ⇒mkClitics "n" "tu" n;
34       -- about 20−30 more table rows
35       ...
36       };
37    Adjective : Type = {s : Str  ;  position  : Position ;
38     isProper  : Bool;  isPrep : Bool};
39   mkAdjective: Str  → Position  → Bool → Bool →
40     Adjective  = \ a  , pos, isProper , isPrep  →
41      { s = a  ;  position  = pos  ;  isPre  = True;
42        isProper  = isProper ;  isPrep  = isPrep };
43   Adverb : Type = {s  : Str ;  agr  : AgrExist} ;
44   mkAdv : Str  → AgrExist  → Adverb =
45     \ str ,  agr  →{s=str ;  agr=agr};
46   NounPhrase : Type = {s  :Case ⇒ Str ;
47     agr  : Agreement};
48   VerbPhrase  : Type = {  s: Str ;  pres  : Str ;
49     perf  : Str ;  isPresBlank  : Bool;
50     isPerfBlank  : Bool; isRegular  : Bool;
51     comp:Str  ;  comp2:Str;  ap  :  Str ;
52     isCompApStem : Bool; agr  :  AgrExist ;
53     adv: Str ;  containsAdv:  Bool;  adV:Str ;
54     containsAdV:Bool};
55   Clause  : Type = {
56     s  : Str  ;  subjAgr  : Agreement; root  : Str ;
57     pres:  Str ;  perf:  Str ;  isPresBlank  : Bool;
58     isPerfBlank  : Bool;compl :  Str
59   } ;
```

Listing 1: Pseudo Code for Parameter, Record & Table Types & Operations in Resource Module

```
1    --Determiners can be lexical  types  or  Phrasal  Type
2    --especially through  DetQuant,DetQuantOrd
3    Determiner  :  Type = {s  :  Str  ;  s2:  Agreement⇒Str;
4      ntype  :  NounState  ;  num : Number ; pos  :  Position ;
5      doesAgree:  Bool;  firstFieldisEmpty  :  Bool;
6      isQuant : Bool};
7    --  prepositions  sometimes  have  two  kinds ,  near  or  far
8    --  i.e omu or omuri
9    --  Can be genetive
10   Preposition  = {s, other  :  Str ;  isGenPrep  :  Bool};
```

Listing 2: Category linearisation Types in StructuralCgg.gf

	Singular	Plural
Complete	omuntu	abantu
Incomplete	muntu	bantu

Table 3: The possible inflectional forms for the noun "omuntu" meaning person.

4.2. Verbs

In R&R verbal inflection depends on tense,[1] Anteriority[2] (2), Polarity (2), Noun Class of Subject, Direct Object and Indirect Object markers (33 * 6 (Person and Number) each) bringing the total possible number of combinations to 124,198,272 inflections which are impractical to enumerate and cannot be handled by the GF compiler at the moment. Apart from the Subject marker (S), Object and Indirect Markers are optional because their use eliminates the need to mention the direct and indirect object(s) in declarative sentences of R&R. We therefore decided to cater for only Subject markers bringing the number down to 3,168 inflections. We found that this number was still prohibitive to successful compilation of the grammar. In light of the above, it was impossible to design a smart-paradigm for verbs. Our solution to the problem involved building the verb at sentence level by designing smaller tables and morpheme-generating operations in Resource modules of both languages. These operations are simply used when necessary as we dynamically built the verb from the radical up to its full form. The operations are of the form "mkXClitics" and "mkXCliticTable" depicted in lines 25–30 & 31–36 of listing 1. The X stands for agreement concords obtained from Mpairwe and Kahangi (2013a). It should be noted that the verbal template example provided in Table 2 is very trivial because conjugation of the R&R verb *reeba* i.e. "to see" from Universal to Perfective form is easy. You simply replace the final vowel *"a'* with the morpheme *"ire"*. In actual sense there exists thirty-eight rules for converting a verb in the imperative mood to the perfective mood. The rules depend on the number of sylla-

[1]We implemented using GF-specific language-independent tense system consisting of Past, Present, Future and Conditional)

[2]Anteriority is a phenomenon used to model grammatical aspect in a manner universal to all languages. It divides each tense into those in which the action is completed (Anterior) versus lack of completeness (Simultaneous)

bles in the verb (mono-, di- and tri- syllabic among others), the length of the penultimate vowel and the letters composing or modifying the terminal syllable such as ***-sa,-sh-,-za,-zya*** or the semi-vowels ***-w or -y***. This can be encoded as a smart paradigm for verb conjugation but the dictionary already gives the set of terminal letters of the verb that must be replaced with the right perfective ending. For example, the entry for the verb *gyenda* in the R&R dictionary by (Mpairwe and Kahangi, 2013a) is marked by *da-zire* to mean that in order to convert the imperative into perfective, replace the *"da"* in *gyenda* with *"zire"* to form *gyenzire*. We did not cover the full spectrum of grammatical aspects possible apart from those required for the language-independent implementation using the concept of Anteriority in GF-RGL. We aim to provide these aspects in a separate Tense / Aspect system within the GF-RGL as extensions in the future.

4.3. Determiners

In R&R it is impossible to express the definite and indefinite articles as distinct words. However Asiimwe (2007) suggests that definiteness can be expressed morpho-syntactically using the Initial Vowel on the noun and other constituents in the noun phrase. Demonstrative determiners are peculiar in that the word used depends on its position on the spatial dexis (Proximal, Medial and Distal), resulting in three words for each noun class. We chose to implement the former two as standard but leave the third form for implementation as an extension. The determiner agrees with number and noun class of the noun. Determiners can be derived from composition of other lexical types (such as Quantifiers and Numerals) via abstract functions: "DetQuant" and "DetQuantOrd" in the abstract syntax. This implies that these non-constant functions add complexity to the modelling of the determiner. Different determiners may appear either before or after the noun hence the need to have a field to track the position they take in Noun Phrases constructed for example by "DetCN" and "DetNP". For the linearisation category type we used a record within another record (refer to lines 3–6 in listing 2) . The string field in the outer record is for determiners that appear before a noun and do not inflect with the Noun while the table of Agreement to Strings inside the inner record is for demonstrative determiners which agree with the noun. The words "every"meaning *buri* and "much" meaning *-ingi* are examples of determiners that take "Pre" and "Post" positions of a noun. Additionally, we have to track whether the determiner 1) composes with either a Complete or Incomplete noun in the "nounCat" field, and 2) is obtained from one of the composing functions. This example for determiners demonstrates the kind of thinking process involved. This process necessitates redesigning types as one encounters new knowledge about the behaviour of Syntactic categories.

4.4. Adjectives

The two languages have two major kinds of adjectives; those that stand alone as their Indo-European counter parts and adjectival stems that require adjectival prefixes derived from the noun class particle of the noun they qualify (Mpairwe and Kahangi, 2013b). Stand–alone adjectives are of two types, those that require the use of possessive pronouns such as *ya* ("of" in noun class *MU_BA*). Some adjectival stems already exist but a large number can be derived from verbs that bear the same or similar semantic meaning of the adjective in mind. Derivation is done by affixing the conjugated copulative verb "ri" i.e. (Subject Prefix + ri) as a prefix to the verb. An example is "-ri-kutagáta" which is derived from the verb kutagáta (to be warm). Lastly, depending on the adjective, it can either occur before or after the nominal (noun/noun phrase). A summary of this information is given in Table 4 and the linearisation category type for Adjective is given on lines 37–38 in listing 1.

Adjective Type	**Example**
Self-standing	Kaganga (Very Large)
Self-standing (Genitive Prepositional)	kijubwe (green) emotoka ya kijumbwe
Adjectival Stem	-rungi (nice) -kwostya (others)

Table 4: The various forms of the adjectives possible.

4.5. Numerals

We implemented Numerals for R&R by following the abstract syntax designed by Hammarström and Ranta (2004). This abstract syntax attempts to give a general yet prototypical representation of numbers of several languages taken from different parts of the world. Since numbers can be nouns, quantifiers, determiners, adjectives or adverbs, modelling them becomes difficult because we have to track agreement concords attributed to gender. Numerals are inherently nouns since they give names to entities used for counting (Ordinals) and order (cardinals). However, Numerals are also quantifiers of nouns i.e. they give an indication of how much or big other nouns are. Being a noun, each numeral belongs to a noun class and therefore has an initial vowel and a noun class particle. When used in quantification of other nouns, the numeral drops the initial vowel and acquires the prefix of the noun or noun phrase it quantifies. The agreement marker (Noun Prefix) acts as a prefix to the last word of the number. For instance, take the example "two hundred and forty people". The number "two hundred and forty" in R&R is *magana abiri na ana* while the noun phrase "two hundred and forty people" is *abantu magana abiri na ba-ana*. Some numerals can be pluralised while others cannot for example you can have "one 6" (*omu-kanga gumwe*) and "two groups of 6" (*emikanga ebiri*). The counting system is awash with synonyms attributed to the evolution of the language over time and the influence of English. The surface form of numerals depends on whether the numeral is Cardinal or Ordinal. When numerals are used in noun phrases the surface form of the number depends on the number and noun class of the head noun in the noun phrase. Therefore we modelled the numeral using tables to store the numeral with its various inflectional forms while keeping the gender and number information as record fields.

4.6. Phrasal Categories

Phrasal categories are derived from the combination of one or more lexical items. The rules for creating phrasal categories are declared in the abstract syntax as functions that take lexical categories as arguments. In GF-RGL abstract syntax, common nouns, proper nouns and pronouns by themselves can be noun phrases. They can also be formed from the combination of a determiner with a noun. The linearisation category type of the noun phrase (refer to line 46–47 in listing 1) stores all forms of the surface string dependent on case. A record field is used to store the agreement information for the noun contained in the noun phrase. Verb phrases are formed from verbs and their complements. Complements maybe noun phrases, adverbial phrases and adjectival phrases. The number of complements the verb may take are one, two or none. All this complement information is stored using fields in the record for verb phrase. In GF-RGL, the clause type is used as a phrasal category to store information for various components of a sentence i.e. Subject (usually a noun phrase) and Verb Phrase. We modelled the clause using a record structure that stores: the Subject as a string and agreement information to be used at the sentence level for determining the Subject marker located in the verb. At the sentence level, clauses are converted to strings according to tense, polarity and Simultaneity (GF-RGL way of covering aspect in language neutral way) to form actual strings for the sentence. Since we could not carry around big tables from Verb-level to Sentence level, we kept the different agreement concords in table structures that can be called upon when needed. The formation of sentence is perhaps the most complicated because morphemes for tense, aspect, polarity and subject markers within the verb must be determined and placed in their various positions according to the verbal template given in Table 2.

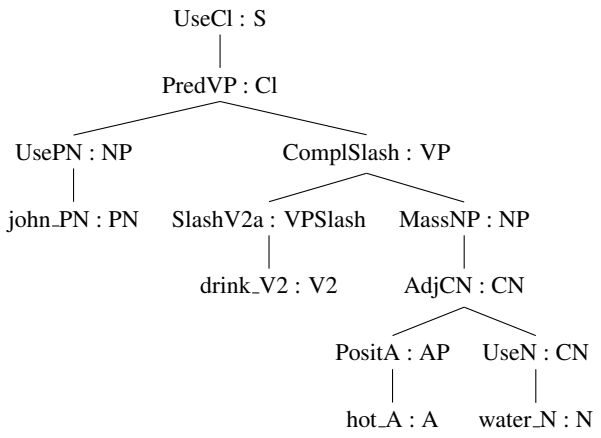

Figure 1: A GF abstract syntax tree generated from parsing "John drunk hot water"

5. An Example and Observations

In this section, we explain how an example GF abstract syntax tree depicted in figure 1 linearises (linearisation is the process of generating strings in a particular language from a parse tree) to Runyankore and Rukiga. The example was generated from parsing the English sentence "John drunk hot water" using GF for the purpose of generating a parse tree. Actually GF generated three parse trees but we chose just one of them for which we had all syntax functions implemented for both Runyankore and Rukiga in the RGL. GF generates *Yohana anywire amáàîzi aga kwotsya* and *Yohana azáànywire amáàîzi aga kwosya* as Runyankore and Rukiga linearisations for the abstract tree in figure 1. The nodes of the parse tree are GF-RGL syntax functions and their return types (the linearisation categories). When we linearised this abstract tree to English, Runyankore and Rukiga, we obtained concrete syntax trees for the languages in figures 2 and 3 for English and Runyankore respectively. We have left out the tree for Rukiga because it it is similar to that of Runyankore. The only difference is the spelling of "hot" being *kwosya* for Rukiga as opposed to *kwotsya* for Runyankore. The English concrete syntax tree is straight forward with each word from the sentence linearised from the leaves of the abstract syntax tree in figure 1. For the two R&R, a special bind symbol "&+" is used for concatenation i.e combining morphemes without spaces. Translation via GF is direct translation so the translations obtained may not be what a native speaker would use. However, they are grammatical i.e. they follow the syntax rules of the language. We made two observations about Runyankore & Rukiga from the parse trees: 1) concrete syntax trees are similar for the two languages and 2) the parse trees of Runyankore and Rukiga are more complicated in relation to English. The explanation for the first observation is that the grammar of the two languages are nearly identical with the exception of a few grammar rules and lexical items. The second observation stems from the fact that the languages are agglutinating resulting in several morphemes within a given word that are connected with grammatical features such as tense, aspect, mood, grammatical number, Person and noun classes. The function "play_V" respon-

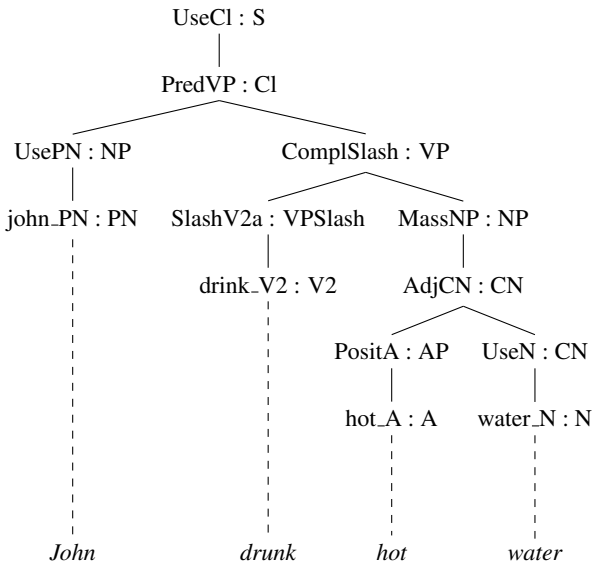

Figure 2: A GF concrete syntax tree generated from linearising the parse tree of 1 into English

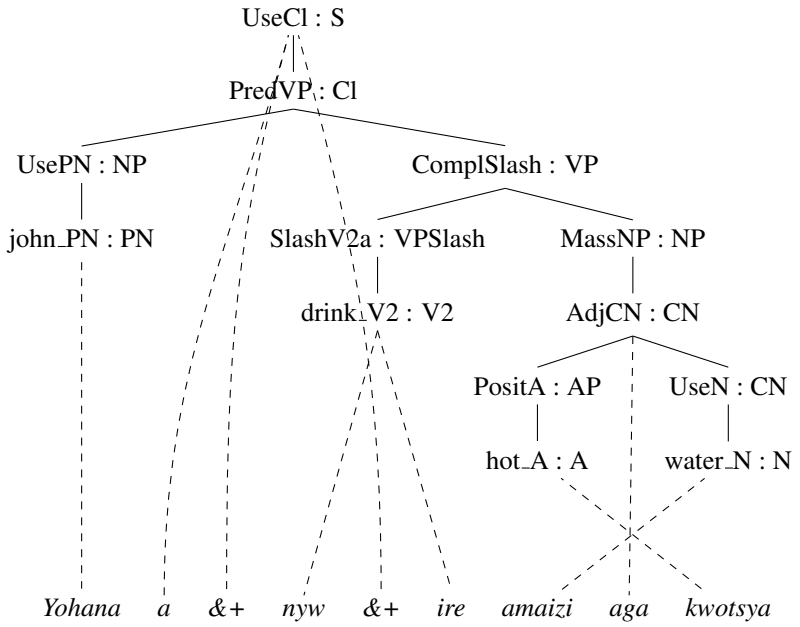

Figure 3: A GF concrete syntax tree generated from linearising the parse tree of 1 into Runyankore

sible for linearisation of the verb play cannot have all its forms conjugated in a paradigm because of the millions of possibilities as discussed already in section 4.2., hence we decided to handle it at sentence level. While implementing the grammar of these languages, we also observed that Runyankore has more resources in terms of grammar books and dictionaries with most books concentrating on Runyankore as opposed to Rukiga.

6. Discussion

During the implementation of GF-RGL for Runyankore and Rukiga we observed that the difference between these languages lies only in a few lexical items. We therefore implemented Rukiga and reused its grammar for the implementation of Runyankore. The only changes we had to make were lexical items specific to Runyankore i.e those not shared by the two languages and a few rules for tenses. In total, we have implemented 290 abstract functions of which, 167 are lexical rules while 123 are phrasal rules. The missing rules consist of 400 lexical and 280 phrasal rules. We computed the 50 most used functions on wordnet and found that we implemented 43 of those functions which is not bad coverage. We plan to perform a proper evaluation in the future after compiling huge lexica and building application grammars for language-learning applications based on this GF-RGL. We simplified the verbal template by ignoring the use of the direct and indirect Object-markers because use of such markers would require anaphoric resolution, which occurs at the discourse rather than the syntactic level. GF-RGL's ability to do multilingual translation based on its universal abstract syntax prevented us from implementing all forms of lexical and syntactic categories because it would break multilingual translation. However, GF-RGL is flexible enough to allow the grammarian to implement language specific features as extensions, which we have done for structural words and intend to do for other syntactic categories. During the development of the grammar, we used regression tests by repeated linearisation of GF abstract syntax trees to English, Runyankore and Rukiga to check for grammatical correctness and ensure our changes did not break existing functions. Phonological conditioning is a particular problem for R&R which we have managed to solve only in our smart noun paradigm. A global solution would require development of morphological analyser and generator for the two languages.

7. Conclusion and Future Work

In this paper, we have described our work on the development and implementation of computational resource grammars for Runyankore & Rukiga Languages. We have succeeded in the modelling and implementation of the morphology and syntax of the languages using GF. The result has been a resource grammar for each language that together have been made freely made available under an open-source licence on GF's Github. In the near future we plan to: complete the Resource Grammar Libraries for the two languages by including language-specific tense and aspectual forms for verbs packaged as additional modules and development of morphological analysers and generators as efficient tools for handling phonological conditioning. We would also like to collect a corpus on which we shall perform an evaluation of the performance of the resource grammars developed. We are currently compiling a large computational lexicon for the two languages which shall increase the coverage of our lexicon. The increase in lexical coverage improves the quality of end user applications developed using resource grammars. Lastly, we will build application grammars in the domain of Computer-assisted language Learning for teaching learners of the two languages about the mechanics of the grammars of these languages.

8. Acknowledgements

This work was supported by the Sida / BRIGHT Project 317 under the Makerere–Sweden Bilateral Research Programme 2015–2020.

9. Bibliographical References

Asiimwe, A. (2007). Morpho-syntactic patterns in Runyankore-Rukiga. Master's thesis, NTNU – Norwegian University of Science and Technology.

Byamugisha, J., Keet, C. M., and DeRenzi, B. (2016). Bootstrapping a Runyankore CNL from an isiZulu CNL. In Brian Davis, et al., editors, *Controlled Natural Language*, pages 25–36. Springer International Publishing.

Byamugisha, J., Keet, C. M., and DeRenzi, B. (2018). Pluralizing nouns across agglutinating Bantu languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2633–2643. Association for Computational Linguistics.

Détrez, G. and Ranta, A. (2012). Smart paradigms and the predictability and complexity of inflectional morphology. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 645–653, Avignon, France, April. Association for Computational Linguistics.

Dymetman, M., Lux, V., and Ranta, A. (2000). Xml and multilingual document authoring: Convergent trends. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING 00, pages 243–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hammarström, H. and Ranta, A. (2004). Cardinal numerals revisited in gf. In *Workshop On Numerals In The World's Languages*, Leipzig, Germany.

Katushemererwe, F. and Hanneforth, T. (2010a). Finite state methods in morphological analysis of Runyakitara verbs. *Nordic Journal of African Studies*, 19(1):1–22.

Katushemererwe, F. and Hanneforth, T. (2010b). Fsm2 and the morphological analysis of Bantu nouns – first experiences from Runyakitara. *International Journal of Computing and ICT research*, 4(1):58–69.

Kituku, B., Nganga, W., and Muchemi, L. (2019). Towards kikamba computational grammar. *Journal of Data Analysis and Information Processing*, 07(04):26, October.

Kolachina, P. and Ranta, A. (2016). From Abstract Syntax to Universal Dependencies. *Linguistic Issues in Language Technology*, 13(3):1–57.

Lange, H. and Ljunglöf, P. (2018). Putting control into language learning. In *CNL 2018: Sixth International Workshop on Controlled Natural Language*, volume 304 of *Frontiers in Artificial Intelligence and Applications*, pages 61–70, Maynooth, Ireland. IOS Press.

Lange, H. (2018). *Computer-Assisted Language Learning with Grammars. A Case Study on Latin Learning*. Licenciate thesis, University of Gothenburg, Sweden.

Maho, J. F. (2009). NUGL Online: The online version of the New Updated Guthrie List, a referential classification of Bantu languages. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.603.6490`.

Meeussen, A. E. (1967). Bantu grammatical reconstructions. *Africana Linguistica*, 3(1):79–121.

Morris, H. F. and Kirwan, B. E. R. (1972). *A Runyankore grammar, by H. F. Morris and B. E. R. Kirwan*. East African Literature Bureau Nairobi, [rev. ed.] edition.

Mpairwe, Y. and Kahangi, G. (2013a). *Runyankore-Rukiga Dictionary*. Fountain Publishers, Kampala.

Mpairwe, Y. and Kahangi, G. (2013b). *Runyankore-Rukiga Grammar*. Fountain Publishers, Kampla.

Muzale, H. R. T. (1998). *A Reconstruction of the Proto-Rutara Tense / Aspect System*. Ph.D. thesis, Memorial University of Newfoundland, Canada.

Ngángá, W. (2012). Building swahili resource grammars for the grammatical framework. In Diana Santos, et al., editors, *Shall We Play the Festschrift Game? Essays on the Occasion of Lauri Carlson's 60th Birthday*, pages 215–226. Springer Berlin Heidelberg, Berlin, Heidelberg.

Perera, N. and Ranta, A. (2007). Dialogue system localization with the gf resource grammar library. In *Proceedings of the Workshop on Grammar-Based Approaches to Spoken Language Processing*, SLP '07, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pretorius, L., Marais, L., and Berg, A. (2017). A GF miniature resource grammar for Tswana: modelling the proper verb. *Language Resources and Evaluation*, 51(1):159–189.

Ranta, A., Angelov, K., and Hallgren, T. (2010). Tools for multilingual grammar-based translation on the web. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 66–71, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ranta, A. (2009a). GF: A multilingual grammar formalism. *Linguistics and Language Compass*, 3(5):1242–1265.

Ranta, A. (2009b). The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2(1).

Ranta, A. (2011). *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford.

Simons, G. F. and Fennig, Charles, D. (2018). *Ethnologue: Languages of the world*. SIL International, Dallas, Texas, Twenty-first edition. Online version:`http://www.ethnologue.com`.

Taylor, C. V. and Yusuf, M. (2009). *A simplified Runynakore-Rukiga-English DICTIONARY*. Fountain Publishers, Kampala, revised ed edition.

Turyamwomwe, J. (2011). Tense and aspect in Runyankore-Rukiga, linguistic resources and analysis. Master's thesis, NTNU – Norwegian University of Science and Technology.