Fsm2 and the Morphological Analysis of Bantu Nouns – First Experiences from Runyakitara

FRIDAH KATUSHEMERERWE[6]
Faculty of Computing and Informatics Technology
Makerere University

THOMAS HANNEFORTH
Department of Computational Linguistics
Institute of Linguistic, University of Potsdam

Abstract:

This paper describes the implementation of Finite State methods, fsm2 in particular, in automatic analysis of Bantu nouns in one of the under resourced languages, Runyakitara. This is the first effort towards computational analysis of Runyakitara. A detailed description of Runyakitara noun classes and how they were analysed using fsm2 is given. In the current state of developing the system, 80% of Runyakitara nouns are correctly analysed. This is a positive step in confirming the success of fsm2 in the analysis of morphology of Bantu languages.

Key words: Finite-State methods, fsm2, morphological analysis, Bantu languages, Analysis of Runyakitara

IJCIR Reference Format:
Fridah Katushemererwe and Thomas Hanneforth. *Fsm2* and the Morphological Analysis of Bantu Nouns – First Experiences from Runyakitara. International Journal of Computing and ICT Research, Special Issue Vol. 4, No. 1, pp. 58 - 69. http://www.ijcir.org/Special-Issuevolume4-number1/article7.pdf.

1. INTRODUCTION

The need for computational morphology as an input for other text analysis applications is core, but literature on computational morphology for most Bantu languages is still scanty. Morphological analysis of natural languages is a well studied area and finite state methods have already been confirmed to effectively analyze the morphology of natural languages [Karttunen, 2005]. Finite-state technology is considered the preferred model for representing the phonology and morphology of natural languages [Wintner, 2007]. The model has been used to computationally analyse natural languages such as English, German, French, Finnish, Swahili, to mention a few cases [Beesley and Karttunen, 2003]. Most implementations on Bantu languages however, have used *lexc* and *xfst*, [Beesley and Karttunen, 2003], therefore, it is on this basis that the fsm2 was selected to be applied on the morphological analysis of Runyakitara nouns so as to provide another implementation perspective.

Using finite-state methods (*fsm2*), a comprehensive system containing all significant lexemes of Runyakitara nouns was compiled. So far, the Runyakitara noun morphological analyser is a combination of a symbol specification, a noun grammar module and a replacement rule module. The purpose of developing the tool is to provide sharable morphological grammar rules of Runyakitara nouns in an organized framework so that they can be used for other applications. Currently, there are no such rules for

[6] Author's Address: Fridah Katushemererwe, Faculty of Computing and Informatics Technology, Makerere University, fkatushemererwe@gmail.com, katu@arts.mak.ac.ug ; Thomas Hanneforth, Department of Computational Linguistics, Institute of Linguistic, University of Potsdam, Thomas.Hanneforth@uni-potsdam.de, tom@ling.uni-potsdam.de

International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Special Issue, Vol.4, No.1, pp. 58 - 69, October 2010.

Runyakitara, yet if important language applications like spell-checkers are to be developed for the language; a word grammar checker is required.

Although Bantu languages are classified as largely agglutinative and exhibit significant inherent structural similarity, they differ substantially in terms of their phonological features implying that each Bantu language requires an independent morphological analyzer.

This paper focuses on the treatment of nouns of Runyakitara, a Bantu language in Finite-State programming environment. The description of nouns was chosen, because nouns constitute a major word category in Runyakitara and play a major role in syntactic analysis. Secondly, the noun classification system in Runyakitara is computationally interesting.

The remaining sub-sections on section one provide an over-view of Runyakitara language and related research on Bantu morphological analysis. Section two details a Runyakitara noun morphology, while three and four describe the formalization and implementation processes. Section five, gives the conclusion and plans for future work.

1.1 Overview of Runyakitara language

Runyakitara is a name given to the four major dialects found in Western Uganda namely Runyankore-Rukiga and Runyoro-Rutooro [Bernsten, 1997]. Guthrie [1967] classifies the languages as narrow Bantu, of Niger-Congo family, Nyankore-kiga (E.13), and Nyoro-Ganda (E.11). The languages are spoken by approximately six (6) million people in fifteen (15) districts of western Uganda. The four dialects are mutually intelligible to the extent that their lexical similarity is between 64% and 94%. That is why Runyankore-Rukiga use the same standard orthography and so is Runyoro-Rutooro. The sub-dialects of the mentioned languages are spoken in the Democratic Republic of Congo (DRC) and some parts of Tanzania, but in these countries, each dialect has its own name; Ruhaya in Tanzania, and Rutuku in DRC. In Uganda, Runyakitara is a standardized term referring to the four languages named, and it is one of Uganda's major languages. The language is taught at Makerere University (Uganda's oldest and largest institution of learning), in some private universities such as Kabale University, and in some primary schools in Western Uganda. The language has borrowed a lot from English (the official language of Uganda), Luganda and Kiswahili.

The above overview is important, because Runyakitara is a major language that needs research, and more so, lexical items that are borrowed tend to behave differently from original words of the language.

1.2 Previous work on morphological analysis for Bantu languages

Considerable amount of work has been done in utilizing finite state methods for Bantu language processing. Using Xerox Finite State tools, Karttunen [2003] uses a realizational framework to model Lingala verb morphology. This approach focuses on using replacement rules to gradually construct the verb from the root, piece by piece.

The Xerox finite state technology has also been utilized in the development of the analyzer prototype for Zulu (Pretorius & Bosch, 2003). This analysis uses lexc, xfst and replacement rules to account for the morphotactics, morpho-phonology and orthographical issues in Zulu language. To account for long distance dependencies found in Zulu morphology, Pretorius and Bosch use flag diacritics as described in Beesley & Karttunen, [2003].

Work has also been done in Swahili using a language-specific morphological parser [Hurskainen 1992] known as SWATWOL. This parser is a two-level analyser that similarly accounts for morpho-syntax and morpho-phonology of Swahili.

Related to the above is the Swahili language manager [Hurskainen, 2004]. The Swahili language manager, in brief SALAMA, is a storehouse for developing multiple computational applications. It is a computational environment for managing written Swahili language and for developing various kinds of language applications. It comprises the standard Swahili lexicon, a full morphological and morpho-phonological description of Swahili, a rule-based system for solving word-level ambiguities, a rule-based system for tagging text syntactically, a rule-based system for handling idiomatic expressions, proverbs and other non-standard clusters of words and semantic tagging and disambiguation system for defining correct semantic equivalents in English. SALAMA is language specific and it does not necessarily account for the types of problems encountered in Runyakitara.

Muhirwe, [2007] describes a computational analysis of Kinyarwanda morphology. The author applies the Xerox finite state compiler to model Kinyarwanda phonological alternations concentrating on orthographical rules. It is important to note that rules are language dependent. Therefore, the rules for

Kinyarwanda or Kiswahili language are not directly applicable to other Bantu languages although they belong to the same group – Bantu language group.

Finite state methods have also been applied to the analysis of Seswana verb morphology (Pretorius, 2008); tonal marked Kinyarwanda [Muhirwe, 2009; Hurskainen, 2009] and solutions for reduplication in Kinyarwanda [Muhirwe & Trosterud, 2009].

There is considerable work that has been done on specific languages mainly applying Xerox finite state methods in morphological analysis and a number of implementations are successful. However, the fact that Bantu languages are more than five hundred (500) in number means a great fraction is still unaccounted for, and there is evidence that Runyakitara is not yet participating in the literature available. In addition, *fsm2* as a scripting language has not yet been applied or implemented to any of the Bantu languages. This makes a publication on the application of *fsm2* in automatic analysis of Runyakitara nouns unique and relevant.

1.3 Methodology

The design of the system was done in three phases: formalization, implementation, and testing. Formalization involved most of the linguistic investigation required throughout the course of the design. Nouns were extracted from a dictionary, '*Kashoboorozi y'ORunyakitara'* [2007]. Initially, manual coding was done to identify sub-classes on main classes of nouns. Classes that do not have prefixes also had to be identified manually. Also to note is that, for nouns in *Kashoboorozi*, noun class prefixes are not marked on their entries, so manual work was comprehensive.

The core of the system is a grammar implemented in *fsm2* formalism [Hanneforth, 2009]. All the regular aspects of nouns were encoded as regular expressions following quasi context free grammar framework. The replacement rules were encoded as regular expressions and compiled into *fsm2*. The grammar and rules were composed together using a composition operator also catered for in *fsm2*.

When the model was completely implemented, it was tested using the lookup tool, also provided for in fsm2 [Hanneforth, 2009]. Testing was done on a corpus of Runyakitara nouns extracted from a weekly newspaper (*Orumuri*) and a teachers' handbook of *Runyankore-Rukiga* orthography.

2. HIGHLIGHTS ON RUNYAKITARA NOUN MORPHOLOGY

Similar to all Bantu languages, Runyakitara has a noun class system. Demuth [2003] describes Bantu noun classification system as such: they are realized as grammatical morphemes rather than independent lexical items. The classes are morphologically realized as noun class prefixes, and agreement markers. The statement, 'agreement markers' means that nouns function as part of a larger concordial agreement system. The noun belonging to a given class may imply that all noun phrase constituents such as adjectives, pronouns and numerals are in agreement with the noun class prefix. Although some writers say that the semantic productivity of Bantu noun classes has reduced, this may need further research because some Bantu languages are not well documented.

Researchers in Bantu languages agree that that noun class features are determined by grammatical number, semantics, (that is, whether they are human/animal/non-living things); and in other cases arbitrarily [Aikhenvald, 2006; Katamba, 2003].

In Runyakitara, a noun can be analysed as a stem and an affix; the affix is mainly a prefix. Suffixation occurs mainly on derived forms from verbs, adjectives and adverbs. This is done by adding an appropriate class prefix on one hand and by replacing the final stem vowel on the other. Such nouns are treated under their respective classes, for example, a noun *omu-shom-i* (reader) from *ku-shom-a* (to read), is catered for in class 1/2 for humans, and the derivational process in this case is not relevant.

Nouns in Runyakitara are associated with an initial vowel as a pre-prefix to the root or stem. These are **a,** (**aba**ntu) **e**, (**eki**tookye) and **o**, (**omu**ntu) as presented by Ndoleriire & Oriikiriza [1990]. There are rules that govern the occurrence of the initial vowel. If the noun class prefix has the vowel **a**, e.g. **ba**, **ma**, the initial vowel will be **a,** thus, **ama**ta (milk) **aba**kazi (women). When the noun prefix has **i** or -, the initial vowel is **e** for example, **e**k**i**tookye, **e**m**i**ti, etc. The initial vowel is o when the noun class prefix has **u**, **o**m**u**ntu (person), **o**m**u**ti (tree). When a noun is pre-ceded by a preposition such as **omu** (in) **aha** (at), the initial vowel is dropped e.g. *omu muti* (in the tree).

Although Bantu languages have a general noun classification system, each language has its own unique sub-classification system, therefore, the noun classification of Runyakitara is considered as knowledge which should be shared.

Whereas nominal morphology is a well studied area in Bantu, classification systems still lack a detailed account especially when considered for computational analysis. This is part of what has motivated a detailed description and computational analysis of Runyakitara morphological grammar.

2.1 Runyakitara noun classification system

The noun class system used in this analysis has borrowed a lot from Katamba [2003] and Taylor [1985]. Katamba [2003] provides a detailed comparative analysis of different classification systems, singling out the Bleek-Meinhof system, and its revisions, as standard. This has provided important insights for Runyakitara analysis. To cater for Runyakitara needs, Taylor [1985:124] details a classification system of Runyakitara nouns describing 17 classes, but with few or limited sub-classes. The description on a noun class system of Runyakitara given in Ndoleriire & Oriikiriza, [1990], has twenty (20) noun classes. However, this description falls short of a numbering system and a detailed description of sub-classes which belong to either singular or plural. The table below, therefore, provides a detailed description of a Runyakitara noun class system.

Class	**Singular**	**Plural**	**Semantics**	***Example***	**Gloss**	**Usage**
1/2	o-mu-	a-ba	Human	*o-mu-kazi* *a-ba-kazi*	Woman Women	Takes on both singular and plural
1a	o-mu-	-	Names referring to deity	*o-mu-hangi*	Creator	Only singular
1b/2b	-	baa-	Human, kinship	*shwento* *baa-shwento*	Uncle Uncles	Takes on both singular and plural, but no prefix for singular
2a	-	a-ba-	Human, group	*a-ba-ryakamwe*	Group name	Only plural forms
3/4	o-mu	e-mi-	Plants, fruits,	*o-mu-ti/e-mi-ti*	Tree(s)	Both singular & plural
3a	o-mu-	-	Uncountable	*o-mu-isyo*	Breath	Singular only
4a	-	e-mi-	Abstract names	*e-mi-gyendere*	Way of walking?	Only plural
5/6	e-ri-	a-ma-	Some parts of the body	*e-ri-isho/a-ma-isho*	Eye(s)	Both singular & plural
5a	ei-	a-ma-	Miscellaneous	*ei-teeka/a-ma-teeka*	Policies	Both singular & plural
5b	ei-	-	Abstract names	*ei-tetsi*	Pampered?	Only singular
6a	-	a-ma-	Mass nouns	*a-ma-te*	Milk	Only plural
7/8	e-ki-	e-bi-	Objects, misc	*e-ki-ti/e-bi-ti*	Tree (s)	Both singular & plural
7	e-ki-	-	Abstract	*e-ki-niga*	Anger	Only singular
8	-	e-bi-	Mass nouns	*e-bi-bembe*	Leprosy	Plural only
9/10	en-	en-	Animals and borrowed words	*e-nte*	Cow(s)	Singular and plural
9	-	-	borrowed words, derived words	*ebahaasa*	Envelope (s)	Singular & plural
10	-	-	borrowed words	*bwino*	Ink	Singular & plural
11/10	o-ru-	en-	Insects, plants miscellaneous	*o-ru-shozi*	Mountain(s)	Singular & plural
12/14	a-ka-	o-bu-	Small items, miscellaneous	*a-ka-buuza*	Question mark(?)	Singular & plural
12	-aka-	-	Abstract nouns	*a-ka-bi*	Danger	Abstract
14	-	o-bu-	abstract nouns	*o-bu-cureezi*	To be humble	Abstract
13	-	o-tu-	Abstract and diminutives	*o-tu-ro*	Sleep	Abstract

Class	Singular	Plural	Semantics	*Example*	Gloss	Usage
15/6	o-ku-	a-ma-	Some body parts	*o-ku-guru/amaguru*	Leg(s)	Singular & plural
16	aha-	-	Location	*aha-kaanyima*	Behind the house	Singular
17	oku-	-	Location	*oku-zimu*	Underground	Singular
18	omu-	-	Location	*omu-nda*	In the stomach	Singular
20/21	o-gu-	a-ga-	derogatory	*o-gu-kazi/a-ga-kazi*	Bad/ugly woman	Singular & plural

Table 1: Noun classification system of Runyakitara

There are generally twenty noun classes in Runyakitara although only eighteen are in use because the two are derogatory, and, tend to be ignored especially in written contexts. Most of the classes are paired in singular and plural, however, there are exceptional cases where a class is in either singular or plural as already illustrated. The status of either singular or plural on the table is illustrated by a null prefix in either case.

Also worth noting is that, some Runyakitara nouns do not take on affixes, but will still belong to their semantic classes, for example, *taata* (Dad) in class one and *ebaafu* (basin) in class 9 do not have prefixes and suffixes. The class where such nouns belong is determined by following the concordial agreement markers on the noun constituents like verbs or adjectives, for example, **e**baafu **e**yangye n'**e**yera (my basin is clean).

Derivation is productive in Runyakitara where nouns are derived from verbs, adjectives and adverbs. This process is done by adding an appropriate class prefix, and by replacing the final stem vowel, for example o-ku-ega (to study), when derived becomes o-mu-egi (student). Such nouns are treated under their respective classes determined by the prefixes.

Compound nouns are also productive in Runyakitara. Compound nouns are a result of combining two or more words of different meanings to form one word with a single meaning. The combinations are mainly between noun and noun, verb and noun, and noun and adjective. Such nouns are treated basing on the prefix of the first noun, but most are in class nine which is open to new words.

Reduplicated nouns are rare in Runyakitara, although they can occur in abusive speech, for example, omuntuntu (not-worthy a person). Runyakitara has catered for those reduplicated forms of nouns that are core to the language.

3. FORMALIZATION

Given the above highlights on Runyakitara noun morphology, quasi context-free grammar, specifically employing the simple substitution approach proposed by Mohri & Sproat [1996] is preferred as the appropriate model for Runyakitara morphotactics because:

- Rules to constrain the order of morphemes are easily written and can output strings.
- Noun classes with their semantic roles can easily be catered for in quasi context-free grammar.

Formally a context free grammar is represented as follows:

- **Context free grammar: G = [T, N, S, R]**
  - **T, a set of terminal symbols**
  - **N, a set of non-terminal symbols**
  - **S, a start symbol**
  - **R, a set of production rules of the form:**
    - **N → T = replace N by T**

Modeling Runyakitara nouns using the above approach can be as below:

Non-terminal symbols: [N] → [NP] [NR]

Terminal symbols [NP] → (omu|mu)

[NR] → ntu

Where N, - noun; NP - noun prefix and NR - noun root.

The question now is whether one writes rules for each and every noun root. That would not be feasible and the solution is to categorize noun roots according to their classification scheme. As a result, the categorized roots which belong to the known noun classes of Runyakitara were labeled with an abstract class. For example, class 1-2 was labeled PEOPLE, so that all roots that belong to that class are given a symbol representing roots. The PEOPLE symbol class will then be substituted into grammar.

Below is a table detailing the symbols given to different classes:

Class	**Prefix**	**Semantics**	**SYMBOL FOR COMPUTATIONAL PURPOSES**
1/2	omu-aba	People	PEAPLE
1a	omu	Creator	CREATOR
1b/2b	baa	Kinship	KINSHIP
2a	aba	Group	GROUP
3/4	omu-emi	Plants	PLANT
3a	omu	uncountable	UNCOUNTABLE
4a	emi	Abstract	ABSTRACT4
5/6	eri-ama	Miscellaneous	MISC
5a	ei-ama	Some Body parts	BODY
5b	ei	Seasons	SEASONS
6a	ama	Mass	MASS
7/8	eki-ebi	Objects	OBJECTS
7	eki	Abstracts	ABSTRACT7
8	ebi	Mass nouns	MASS8
9/10	en-en	Animals	ANIMALS
9	-	Abstract nouns	ABSTRACT9
10	-	Mass nouns	MASS10
11/10	oru-en	Insects	INSECTS
12/14	aka-obu	Diminutives	ABST12
12	aka	Small and tinny	SMALL
14	obu	Abstract	ABSTRACT
13	otu	Mass nouns	MASS13
15/6	oku	Body parts	BPARTS
16	aha	Locative	LOCATION
17	oku	Locative	LOCA1
18	omu	Locative	LOCA2

Table 2: Classes and symbols representing roots

When the symbols representing roots are incorporated into the grammar, it looks like the extract below:

Non-terminal Symbols
[NOUN] --> [NOUN_PREF1][NOUN_ROOT1]
[NOUN] --> [NOUN_PREF1][NOUN_ROOT1A]
Terminal Symbols
[NOUN_PREF1] --> (omu|mu) [NOUN_PREF_1S 1s=npref1s]
[NOUN_PREF2] --> (aba|ba) [NOUN_PREF_2P 2p=npref2p]

[NOUN_ROOT1] --> [PEOPLE] [NOUN_ROOT_PS Ps=class1]
[NOUN_ROOT1A] --> [CREATOR] [NOUN_ROOT_1SI 1Si=singular1]
Runyakitara noun grammar extract

4. IMPLEMENTATION

The grammar is implemented using *fsm2* [Hanneforth, 2009], a scripting language within the framework of finite state technology. Finite-state technology is considered the preferred model for representing the phonology and morphology of natural languages [Wintner, 2007]. The model has been used to computationally analyse natural languages such as English, German, French, Finnish, Swahili, to mention a few cases [Beesley and Karttunen, 2003], and its main advantage is that it is bidirectional – it works for both analysis and generation. It is on this basis that the technology was selected to be applied on the morphological grammatical analysis of the Runyakitara noun.

Fsm2 was chosen as a resource tool for a morphological grammar of Runyakitara nouns because of a number of reasons: i) it supports a full-set of algebraic operations defined on both unweighted and weighted finite state automata and weighted finite state transducers [Hanneforth, 2009]. Algebraic operations are useful to design complex morphological analyzers in a modular way. ii) *fsm2* supports a number of equivalence transformations which change or optimize the topology of a weighted automation without changing its weighted language or relation, that is an automata can be minimized, determinized, optimized etc; iii) *fsm2* uses symbol signatures which map symbols to numbers that are internally reconised by the automata. Symbol signatures are useful in language modeling since every word in a language constitutes an alphabet symbol, and a task of a developer is to define symbols that represent morphemes and their categories. iv) *fsm2* provides an efficient way of compiling morphological grammars where the co-occurrence of roots and inflectional affixes common in Runyakitara is easily accounted for.

Fsm2 is able to load lexicons, grammars and replace rules defined by the morphology developer. It is able to automatically transform rules into transducers.

4.1 Application to Runyakitara nouns

The noun morphological system is built on a modular basis comprising a special symbol module/file, a noun grammar module and a replacement rule module. The three are composed together, and the result is a single finite state transducer.

The following diagram demonstrates the overall architecture of a noun morphological system:

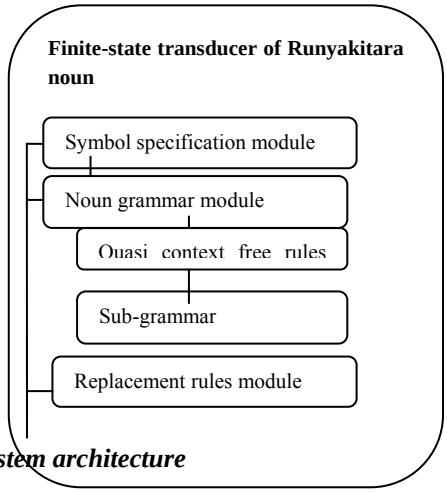

Figure 1: Noun morphological system architecture

4.1.1 A symbol specification module provides a unique mapping from user-defined symbols which are letters, categories, to integers (numbers) which are used internally by the automata and the operations. A symbol signature maps symbols to their internal integer representation on a one-to-one basis to allow symbolic computation [Hanneforth, 2009]. A symbol specification module for a noun is loaded first in *fsm2* before any other file is loaded.

4.1.2 Noun grammar module

The grammar module consists of a sub-set of quasi context-free rules accounting for concatenative nature of Runyakitara noun morphology. The grammar contains a large number of rules, but we present a sample, exemplifying the principles underlying the overall organization of the grammar:

Non-terminal Symbols

[NOUN] → [NP1S] [NROOT1] # omu-ntu (person)
[NOUN] → [NP2P] [NROOT1] # aba-ntu (people)
[NOUN] → [NP3S] [NROOT3] # omu-ti (tree)
[NOUN] → [NP4P] [NROOT3] # emi-ti (trees)

Terminal symbols:

[NP1S] → omu
[NP2P] → aba
[NROOT1] → ntu
[NP3S] → omu
[NP4P] → emi
[NROOT3] → ti

An extract of a noun context free grammar

Notes
[NP1S] – Noun prefix class 1 singular
[NP2P] – Noun prefix class 2 plural
[NP3S] – Noun prefix class 3 singular
[NROOT] – Noun root 1
[NROOT3] – Noun root for class 3
[NP4P] – Noun prefix class 4 plural

The above rules cater for prefixes and a root, and are able to give an output of one root at a time. Since writing rules for each root is not feasible, *fsm2* provides two possibilities of catering for roots when developing a morphological analyzer:

a) **Include statement**. This works in the sense that one writes a sub-grammar containing roots (either verb or noun) preferably in a separate file, then includes the roots into the grammar of prefixes and suffixes using the '#include' command. For example, (i) is a grammar for class one nouns:

(i) [NOUN] → [NP1S][NROOT1]
[NP1S] → omu
#Include [NROOT1]

[NROOT1] → ntu
[NROOT1] → shaija
[NROOT1] → kazi
[NROOT1] → gyenyi

#Include VROOT1 will include *ntu, shaija, kazi*, and *gyenyi* noun roots into a grammar of 'noun prefix 1 singular and noun root1.' The above grammar is catering for the following nouns: omu-ntu (person), *omu-shaija* (man), *omu-kazi (woman)* and *omu-gyenyi* (visitor).
This is the approach selected to implement Runyakitara nouns because it is easy to implement.

b) The second approach is to write lexicons for each class of noun roots, then use a substitution approach to include them in the grammar. Roots, already categorized into noun classes are catered for by lexicons, that is, lexicons are built for each noun

class, compiled into Finite State machines, assigned to nonterminal symbols which in turn are replaced by the corresponding Finite State Machines in the Finite State Machine representing the main noun grammar. The substitution operation supported by *fsm2* is able to substitute the roots into the grammar. Therefore, NROOT, as illustrated above, can stand in for many roots of various noun classes.

4.1.3 Morphotactics

The output from the noun context free grammar is still a set of morpheme concatenations forming strings but some are still abstract concatenations (morphotactics) without proper phonological and orthographical representation. The following can represent a sample of an output of a Runyakitara noun grammar from *fsm2:*

omuegi : omu[NOUN_PREF_1S 1s=npref1s]egi[NOUN_ROOT_PS Ps=class1]
omuegizo : omu[NOUN_PREF_3S 3s=npref3s]egizo[NOUN_ROOT_3SI 3Si=singular3]
omuegoojooro : omu[NOUN_PREF_3S 3s=npref3s]egoojooro[NOUN_ROOT_3SI 3Si=singular3]
omueguzi : omu[NOUN_PREF_1S 1s=npref1s]eguzi[NOUN_ROOT_PS Ps=class1]

Output extract from a noun grammatical system

The above four examples: **omuegi,** omuegizo, omuegoojooro and **omueguzi** are valid morphemes in Runyakitara, representing correct grammatical information, but are not correctly spelt and well pronounced words. The grammatical forms are **omwegi**, **omwegizo, omwegoojooro and omweguzi**. This calls for a change of u to w in all cases. Such and many cases of phonological and orthographical nature have been catered for by replacement rules.

4.1.4 Replacement rules

Rules here cover morpho-phonological and orthographical occurrence. These phenomena are catered for by replace rules. Replacement rules are compiled into finite-state automata. *Fsm2* provides for conditional and unconditional replace rules. An expression:

α -> β / γ_δ

states that: replace alpha by beta whenever alpha occurs in the context of lambda on the left and delta on the right can be manipulated by *fsm2*. Some examples of replacement rules that were included to account for morpho-phonological and orthographical processes are indicated below:

a) u -> w / m_ (a | o | i)

The above rule means: replace **u** by **w**, whenever **u** occurs between **b** and **a** or **o** or **i**. This kind of rule will change *om**u**-egi* to *om**w**egi*, *om**u**-egizo to om**w**egizo, omu-egoojooro to om**w**egojooro* and *om**u**-eguzi* to *om**w**eguzi;* all are well formed Runyakitara words.

A sub-set of replacement rules for Runyakitara nouns was developed following the above framework. The rules in this category are able to delete, substitute, and insert symbols in the string as long as the context is clearly defined.

The grammar transducer, and the context rule transducer are composed together and the result is a single transducer outputting grammatical Runyakitara verbs.

5. GRAMMATICAL OUTPUT

The output of a verb grammar includes morphemes, their categories and features. The following is a sample output of a noun morphological system:

abaakiizi : aba[NOUN_PREF_2P 2p=npref2p]akiizi[NOUN_ROOT_PS Ps=class1]
abaambari : aba[NOUN_PREF_2P 2p=npref2p]ambari[NOUN_ROOT_PS Ps=class1]
abaambuzi : aba[NOUN_PREF_2P 2p=npref2p]ambuzi[NOUN_ROOT_PS Ps=class1]
abaami : aba[NOUN_PREF_2P 2p=npref2p]ami[NOUN_ROOT_PS Ps=class1]
byetengo : bi[NOUN_PREF_8P 8s=npref8p]etengo[NOUN_ROOT_8PL 8Pl=plural8]
byevugo : bi[NOUN_PREF_8P 8s=npref8p]evugo[NOUN_ROOT_IT It=class7]
byeyariro : bi[NOUN_PREF_8P 8s=npref8p]eyariro[NOUN_ROOT_8PL 8Pl=plural8]
byeyemekye : bi[NOUN_PREF_8P 8s=npref8p]eyemekye[NOUN_ROOT_IT It=class7]
byeyera : bi[NOUN_PREF_8P 8s=npref8p]eyera[NOUN_ROOT_8PL 8Pl=plural8]
byeyerezo : bi[NOUN_PREF_8P 8s=npref8p]eyerezo[NOUN_ROOT_IT It=class7]

From the above output, the information on a noun is captured. Taking the first noun as an example, aba is captured as a noun prefix for class two and it is a plural marker. Akiizi is a noun root for people and it is in singular form.

Interestingly, there is a noun *abaami* and *baami* all meaning the same noun but used in different circumstances. As already mentioned, nouns which do not have prefixes like *baami* (chiefs/men) are preceded by a preposition.

6. TESTING

Testing is one of the complex tasks in morphological analyzer development [Beesley and Karttunen, 2003], therefore it needs a lot of care and patience. One of the important aspects of *fsm2* is the testing functionality to aid developers test and debug morphological analyzers. The *fsm2* testing functionality can be described as:

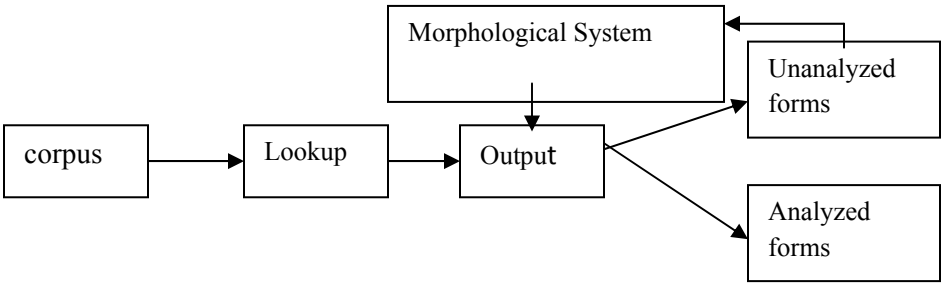

Figure 2: Testing process in fsm2

Applied to Runyakitara, Runyankore-Rukiga nouns were extracted from the weekly newspaper (*Orumuri*) and a Runyakore orthography reference book, [Taylor, 1957]. That constituted our testing raw material. Using the lookup operation provided by *fsm2*, the words were looked up in the analyzer and the result was two files: one of analysed forms and another of unanalyzed forms. The unanalyzed forms were re-examined for further consideration into the noun morphological system.

The following table provides the results:

Corpus (nouns)	**Analyzed forms**	**Percentage (recall)**	**Unanalyzed forms**	**Percentage**	**Correctly analyzed**	**Precision**
5599	4472	80	1127	20	4472	80%

Table 3: results

The above results indicate that the noun system of Runyakitara in development has so far registered success by analyzing 80% of running text. The 1127 forms which were not analysed were not yet included in the system. All the 4472 strings were correctly analysed, both recall and precision are 80%. This is a positive remark on the ability of fsm2 to analyse noun morphology of Runyakitara. During the debugging process (which is our next step) the 1127 unanalysed forms will be processed and included into the noun grammar.

6.1 Applications of Runyakitara noun system

The output which the noun system of Runyakitara generates can be used as an input for other applications such as:

- spell checker of Runyankore-Rukiga
- dictionary since the system can output lemmas
- syntax analyser of Runyakitara
- language learning system for vocabulary and grammar depending on how it can be developed

7. CONCLUSION AND FUTURE RESEARCH

This work demonstrates the application of finite state approach in the analysis of Runyakitara noun morphology. Language specific knowledge and insight have been applied to classify and describe the morphological structure of the language, and quasi context-free and replace rules have been written to account for grammatical nouns of Runyakitara.

The research results presented above describe the first efforts aimed at building a morphological analyser of Runyakitara, a Bantu language with limited electronic resources. The system results from a combination of Item-and-arrangement and Item-and-Process models as proposed by Hockett, [1954; 1959] and it is evident that the models are applicable for Runyakitara morphology.

Specifically, this work demonstrates:

a) the first computational description of the orthography of the Runyakitara nouns
b) proof that the fsm2-inspired approach (context free grammar plus Replacement rules) is applicable to a morphologically complex Bantu language, Runyakitara
c) a computational framework of the noun classification system of Runyakitara. This does not exist in any Runyakitara text book, but was devised in this research for computational purposes.

Future research: The entire plan for this research is to cater for all Runyakitara word categories to be a analysed by *fsm2*. The overall aim is to develop a comprehensive morphological analyser of Runyakitara. The morphological analyzer will be an input for many other planned applications such as learning systems and machine translation.

8. ACKNOWLEDGMENT

The authors would like to acknowledge the continued support and expert advice of Prof. Arvi Hurskainen, of Helsinki, Finland and Prof. John Nerbonne of the University of Groningen, The Netherlands.

This paper is a result of the six month visiting research period in Germany, supported by DAAD (The German Academic Exchange Service). We are grateful.

9. REFERENCES

BEESLEY F. K. AND KARTTUNEN L. 2003 *Finite state morphology* CLSI Publications

BERNSTEN JAN 1998. Runyakitara, Uganda's 'New' Language *Journal of multilingual and multicultural development* vol. 19, No. 2

B□GEL T, ET AL. 2007. Developing a Finite-State morphological analyser for Urdu and Hindi

DEMUTH, K., 2000. Bantu noun class systems: Loan word and acquisition evidence of semantic productivity *in G. Senft ed., Classification Systems. Cambridge University Press*

ELWELL, R. 2006. Finite state methods for Bantu verb morphology *Proceedings of the Texas Linguistics Society X, Austin.*

ESHTON, E. O. 1937. The structure of Bantu language with specific reference to Swahili

GUTHRIE, M., 1967. *An introduction to the comparative linguistics and the pre-history of Bantu languages* Gregg International Publishers ltd.

HURSKAINEN, A. 1992). A two-level computer formalism for the analysis of Bantu morphology: an application to Swahili *Nordic Journal of African Studies*, vol. 1 No. 1 pp. 87-119

HANNEFORTH, T. (2009). *Fsm2* – A scripting language for creating weighted finite-state morphologies in *Mahlow C. & Piotrowski (eds.) State of the Art in Computational Morphology (pp. 48-63)* Heidelberg: Springer Berlin.

HURSKAINEN, A. 1996. Disambiguation of morphological analysis in Bantu languages *Proceedings of the 16th Conference on Computational Linguistics, Capenhagen, Denmark*

HURSKAINEN, A. 2004. Swahili language manager: a storehouse for developing multiple computational applications *Nordic Journal of African Studies*, vol. 13 No. 3 pp. 363-397

KARTTUNEN, L., 2003. Computing with Realizational Morphology. *Computational Linguistics and Intelligent Text Processing.* Alexander Gulbekh (ed.), Lecture Notes in Computer Science, 2588: 205-216.

KARTTUNEN, L. & BEESLEY, K. R., (2005). Twenty-five years of Finite-State Morphology. In inquiries into words, a festschrift for Kimmo Koskenniemi on his 60th Birthday, *CSLI Studies in Computational Linguistics.* Stanford CA: CSLI; 2005; 71-83

KIHM A, (2001) What is in a noun: noun classes, gender, and nounness *CNRS, Laboratorire de Linguistique formelle*

KOSKENNIEMI, K. 1984. Two-Level Morphology: a general computational model for award-form recognition and production *Technical Report, University of Helsinki.*

LADEFOGED, P., et al 1972. Language in Uganda Ford Foundation *Language Survey Vol. 1, London, Oxford University Press.*

LAZAROV, M. 2006. Finite-state methods for spelling correction *unpublished BA thesis.*

MOHRI M. & PEREIRA F. C. N. (1996). Dynamic compilation of weighted context-free grammars. AT&T Labs – Research.

MUHIRWE, J. 2007. Computational analysis of Kinyarwanda morphology: the morphological alternations *International Journal of computing and ICT Research*, vol. 1 No. 1 pp. 85-92.

NDOLERIIRE & ORIIKIRIZA (1990) *Runyakitara Studies*. Unpublished manuscript.

PRETORIUS L. AND BOSCH E.S. 2005. Finite-State Computational Morphology: An analyser prototype for Zulu. Machine Translation, vol. 18 No. 3 pp. 195-216.

SHACHAM, D. AND WINTNER, S. 2007. Morphological Disambiguation of Hebrew: a case study in classifier combination *Proceedings of the 2007 joint conference on Empirical Methods in natural language processing and computational natural language learning, Prague, June 2007, pp.439-447.*

TAYLOR, C., 1985. *Descriptive Grammars, Nkore-Kiga* London, Groom Helm.

TINSLEY, J., 2006. Morphological analysis of Spanish using Xerox Finite-State tools.

WINTNER, S., 2007. Finite-State technology as a programming environment *CICLing 2007, LNCS 4394, pp. 97 – 106.*

YLI-JYRA A., 2005. Toward a widely usable finite-state morphology workbench for less studied languages – part 1: Desiderata Nordic *Journal of African Studies, vol. 14 No. 4 pp. 479-491*