Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2019, 7, 250-275
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Towards Kikamba Computational Grammar

Benson Kituku[1], Wanjiku Nganga[2], Lawrence Muchemi[2]

[1]Computer Science Department, Dedan Kimathi University of Technology, Nyeri, Kenya
[2]School of Computing and Informatics, University of Nairobi, Nairobi, Kenya

Email: benson.kituku@dkut.ac.ke

How to cite this paper: Kituku, B., Nganga, W. and Muchemi, L. (2019) Towards Kikamba Computational Grammar. *Journal of Data Analysis and Information Processing*, **7**, 250-275.
https://doi.org/10.4236/jdaip.2019.74015

Received: September 8, 2019
Accepted: October 19, 2019
Published: October 22, 2019

Abstract

The under-resourced Kikamba language has few language technology tools since the more efficient and popular data driven approaches for developing them suffer from data sparseness due to lack of digitized corpora. To address this challenge, we have developed a computational grammar for the Kikamba language within the multilingual Grammatical Framework (GF) toolkit. GF uses the Interlingua rule-based translation approach. To develop the grammar, we used the morphology driven strategy. Therefore, we first developed regular expressions for morphology inflection and thereafter developed the syntax rules. Evaluation of the grammar was done using one hundred sentences in both English and Kikamba languages. The results were an encouraging four n-gram BLEU score of 83.05% and the Position independent error rate (PER) of 10.96%. Finally, we have made a contribution to the language technology resources for Kikamba including multilingual machine translation, a morphology analyzer, a computational grammar which provides a platform for development of multilingual applications and the ability to generate a variety of bilingual corpora for Kikamba for all languages currently defined in GF, making it easier to experiment with data driven approaches.

Keywords

Grammar, Morphology, Syntax, Grammatical Framework, Under-Resourced language, Concord, Multilingual, Agglutination, Kikamba

1. Introduction

The commonly used data driven approaches for developing natural language processing (NLP) tools are currently unusable with under-resourced languages due to data sparsity and this problem might not be resolved in the near future. There is a high demand for these NLP tools due to the exponential growth of the Internet, which has availed a wealth of information available to people and

coupled with the high penetration rate of connected mobile devices. There is, therefore, an urgent need to devise strategies that can accelerate the development of language technology tools and applications for under-resourced languages so as to enable their speakers to maintain the use of their languages within a digital environment. This paper describes the development of a computational grammar for Kikamba Language, an under-resourced language, using the multilingual Grammatical Framework toolkit.

Guthrie [1] classifies Kikamba language as E55 (Language 5 in group 50 of zone E) in the larger Bantu family and the language commands close to four million speakers. Its grammar is agglutinative, tonal, inflectional and has a noun class system or a class gender (noun prefix and Concord for the noun modifiers) [2] [3] [4]. In addition, its orthography consists of seven vowels and fifteen consonants [5]. In terms of descriptive grammar for Kikamba language, some work is already done, though most of them are not published such as derivational verb morphology [6] [7], noun modification [3], morphosyntax for Kikamba [2] and tonal perspective [8] [9]. Some gaps still exist on these works; for example, the subject marker and negation in verb morphology is only done for class gender which deals with humans only. The concord for possessive pronouns, morph phonological changes in adjectives and verbs, the morphology of compound Nouns and adjectives are yet to be done. With respect to language resource tools, there are only two language tools for this language to the best of our knowledge—these are a Part of Speech tagger and a named entity recognizer [10] [11]. GF has also been used to model language resources for Bantu Languages. Kiswahili language has a partial morphology analyzer [12] while the Tswana Language from South Africa has a mini resource grammar [13]. Hence, no wide-coverage grammar for a Bantu Language has been made in GF so far. Thus, development of the Kikamba Computational Grammar is a significant milestone towards the creation of standard Basic Language Resource Kit (BLARK) [14] since it will result in a Morphological analyzer and multilingual translation using the capability of Grammatical Framework. Secondly, it will be a catalyst to the provision of information and communication technology (ICT) in Kikamba language, thus bridging the digital divide. It will provide a platform for the generation of parallel corpora and treebanks, which are crucial for building NLP tools using data driven approaches. Finally, it is an electronic preservation effort for the Kikamba language so that the Kamba people are not disenfranchised in the global information space.

2. Kikamba Descriptive Grammar

2.1. Morphology

Kikamba language way of forming words from the morphemes is through prefixing and suffixing (agglutination) with the direct influence of noun class system, noun concord and morph phonological transformation. Only a few borrowed words or irregular words deviate from the noun class system prefixing.

Regarding the noun class system, arguments have been advanced whether it should be referred to as gender or noun class. Some consider a pair of singular and plural noun class as gender [15] [16]. This thought is reinforced by Demuth [17] by proposing a noun class as a subset of gender. However, Ibrahim [18] argues that gender or noun class can hold ground since Bantu genders are not inspired by natural sex gender semantics as the case with Indo-European languages. For the purpose of this paper, we shall adopt two pairs of noun classes (singular and plural) forming class gender. **Table 1** lists all noun classes for Kikamba language [2] [3] [4]. The morpheme before the underscore represents the singular noun class while the one after represents the plural noun class and both form the class gender encoded in the third column for use in the GF grammar modeling. We shall discuss the inflection of open and thereafter closed categories.

2.1.1. Noun

The structure of noun morphology consists of obligatory prefix and root plus an optional suffix. The prefix determines the noun class number and we exemplify its usage by Example 1 where the notation "c" means class and the number means noun class number based on **Table 1** (for example c1 means noun class number one), while the root is the radical of the lexical word. The suffix "ni" is used to form a locative noun, which is a case (grammar feature). In the real sense, it is a preposition and a noun combined, for example "at the shop" becomes "dukani" and "on the table" become "mesani". The words "shop" and "table" in Kikamba are "duka" and "mesa" therefore, the preposition is actualized by adding the suffix "ni".

Example 1 Noun structure	
Singular	**Plural**
ki-veti	i-veti
c7-root	c8-root
woman	Women

Table 1. Kikamba noun classes.

Classes (c)	Class number	GF coding
mu_a	1, 2	G1
mu_mi	3, 4	G2
i_ma	5, 6	G3
ki_i	7, 8	G4
ka_tu	12, 13	G5
va_ku	14, 15	G6
n_n	9, 10	G7
u_ma	11, 6	G8
u_n	11, 10	G9
ku_ma	15, 6	G10

2.1.2. Adjective

The adjective describes and modifies a noun and its inflection consists of a prefix (concord) which agrees with the class gender of the noun being modified. In addition, to form the adjective, concatenation of the prefix with the adjective root is done [2] [4]. Example 2 demonstrates the structure of the adjective whereby the adjective prefix is shown by the *noun class* while the radical is shown by *Adjroot*.

Example 2 Adjective structure	
Singular	**Plural**
Mu-ti mu-nini	mi-ti mi-nini
c3-root c3-adjroot	c4-root c4-adjroot
Small tree	Small trees

2.1.3. Verbs

Kikamba language is no exception to the complexity of verb morphology in Bantu languages. Its declension involves several morphemes (several prefixes, root, extensional suffix and final vowel which represent mood) plus some grammar features such as person, number, class gender, tense, polarity, etc. **Table 2** describes all the morphemes used in verb inflection [2] [7]. The object marker, infinitive and extension suffix are not obligatory while in some cases of negative polarity, the subject marker and negation marker are fused together to form one morpheme. Importantly, the focus and negative marker do not co-exist. Finally, the morphemes of verbs embody all the constituents needed to make a sentence, hence the reason a verb can act in place of a sentence. Examples 3 - 6 demonstrate this principle.

Table 2. Architecture of verbs.

Architecture	Morpheme	Kikamba
Prefixes	Focus	"ni"
	Negation	as per class
	Subject marker	as per class and person
	Tense/Aspect	As per tense
	Object marker	as per class and person
	Infinitive	"Ku"
Root		Root
Extension	Applicative	"i'
Suffix	Causative	"ithy"
	Passive	"w"
	Reversive	"u"
	Reciprocal	"an"
Final vowel		"a/e"

Tense

Reichenbach [19] states point of the speech, point of reference and point of the event in relation to time bases for tenses and time is based from speech point [7]. The coincidence of the three points results in the present tense. When the speech point is after the other two points, then past tense occurs. Future tense occurs when the speech point is before other points. Finally, when the reference time proceeds event time, the resultant is perfect tense. The Aspect gives a view of the action of the verb such as beginning, continuing or ended [7]. Most of the time, tense and aspect are combined together in Kikamba languages. Several tenses exist in Kikamba Language [2] [7]. Here we shall exemplify present, future, past and perfect tenses. The following notations are used: *Fs* for focus, *Neg* for negation, *Agr* for the subject marker, *root* for the root, *Tns* for tense, *Asp* for aspect and *Fw* for the final vowel.

The morpheme "ka" marks future tense also referred to as indefinite future tense. The tense morpheme is in-between the subject marker and the root as exemplified in Example 3. Kikamba language has a remote future tense, constructed by concatenating prefix "ni" to the future tense, e.g., using the case of Example 3 we will have "niakakoma", "Gloss", "he will sleep".

Example 3 Future tense	
Positive	**Negative**
Akakoma	Ndakakoma
a *Agr* ka *Tns* kom *Root* a *Fw*	Nda (*Agr* & *Neg*) ka *Tns* kom *Root* a *Fw*
He will sleep	He won't sleep

Past tense is marked by final vowels morpheme "ie" which mark tense though affected by the phonological rule and uses infix "na" to mark aspect [2] [7]. On negative polarity, the infix "nee" is used as exemplified in Example 4.

Example 4 Past tense	
Positive	**Negative**
Nimanakomie	Matineekoma
Ni *fs* ma *Agr* na *as* kom *Root* ie (*Fw* & *tns*)	Ma *Agr* ti *Neg* nee i*nfix* kom *Root* a *Fw*
they slept	They didn't sleep

Present tense, in some cases referred to as present indefinite tense or habitual tense depending on usage, is marked by aspect vowel "a" [7] as exemplified by Example 5.

Example 5 Present Tense	
Positive	**Negative**
Nimakomaa	Maikomaa

Continued

Ni *fs* ma *Agr* Kom *Root* a *Asp* a *Fw*	Mai *Agr* & *Neg* Kom *Root* a *Asp* a *Fw*
they sleep	they don't sleep

Finally, the Perfect tense on positive polarity is not marked by any morpheme though, in the negative, it is marked by morpheme "na" as illustrated in Example 6.

Example 6 Perfect Tense	
Positive	**Negative**
Nitwakoma	Tuinakoma
Ni *fs* twa *Agr* kom *Root* a *Fw*	tui *Agr* & *Neg* na *Tns* kom *Root* a *Fw*
We have slept	we haven't slept

2.1.4. Closed Categories

The demonstrative, a noun modifier which shows how far the object(s) is/are from the speaker and unlike Indo-Europeans languages which have demonstrative strings for near and distant. Kikamba language has an extra string for the aforementioned [2] [3]. Demonstrative inflect for the variable features of class gender and number.

Personal pronouns in Kikamba language stand for absent nouns and in GF they are modeled as noun phrases and therefore have a string and enforce agreement (person, class gender and number). The possessive pronoun, a noun modifier depicting ownership and its architecture consist of a prefix dependent on class, gender and number [3] as exemplified in Example 7 and a root.

Example 7 Pronoun structure	
Singular	**Plural**
Mu-ti wa-kwa	Mi-ti ya-kwa
c3-root c3prefix-*Poss-root*	c4-root c4prefix-*Poss-root*
My tree	My trees

For the preposition, through elicitation, it was noted that the strings for some prepositions have variable features of the class, gender and number for example "of", while most of them do not inflect. In addition, some prepositions are fused into the noun as demonstrated in Example 8, resulting in the locative noun. Cardinal and ordinal numerals can be expressed in words or digits. The cardinal numerals, when expressed in words for the cases of one to five behave like adjectives and take a concord agreement while the rest are independent of the class gender [3]. Ordinal numbers consist of two strings: the preposition "of" and string both dependent on class gender and singular number. Finally, the adverbs do not inflect and there are no articles in Kikamba languages.

Example 8 Preposition fusion			
the pen of John was on the chair			
Kiandiki kya	Yoana	kyai	ki vila-ni
c7-root c7 "of" prefix	Proper Noun c1	to be	c7-root-Loc prefix

2.2. Syntax

The main topology for the Kikamba language sentence is subject-verb-object (SVO) [2] [7] whereby the subject is a noun phrase, followed verb phrase. The verb phrase is a combination of the verb phrase and object complement which can be a verb phrase, noun phrase, etc. The presence of the object is influenced by the verb valence (univalent, divalent and trivalent). For example, for the univalent verb, the topology becomes SV because the one place verb does not require arguments. The syntactic agreement is via concord agreement within the lexical items mainly influenced by the class gender of the noun [2] [3].

Noun phrases are made of a noun and its modifiers which include an adjective (Adj), determiner (Det), both possessive (poss) and demonstrative (dem) and finally numbers (Num). Rugemarila [20] has worked extensively on the structure of noun phrases in Bantu languages and has concluded the structure to be as illustrated below which concurs with one presented by Mbuvi [3] for Kikamba language.

[dem] [Noun] [Det <poss> <dem>] [Num] [Adj]

The structure of a verb phrase is the same as a verb and carries all parameters that are integral to verbs.

3. Translation Approaches

The three main approaches to machine translation are: data driven, rule based and hybrid strategies [21]. The data driven approach, such as neural network models, statistical models, etc. makes use of parallel aligned corpus to make the machine translation possible. It is divided into statistical and example based translations. The rule based approach uses syntax, lexical rules and a lexicon to form a computational grammar based on Chomsky theories [21]. Word-based, transfer and interlingua are the three subcategories of rule based approaches. A grammar formalism determines the architecture of the grammar. The hybrid approach involves using the above approaches together with either a rule based guided hybrid or data driven hybrid translation. In section one, we mentioned Kikamba language being an under resourced language. Thus, very few digital corpora are available, that is why we used the Interlingua rule based translation approach. The Grammatical Framework was chosen because first, its multilingual capability enables the creation of the technology in the different languages already defined in GF. Secondly, separate tecto-grammatical (abstract syntax) and pheno-grammatical (concrete structure) [22] enable faster development since one concentrates on only the concrete syntax of the language been devel-

oped. Finally, it provides a platform where application grammars can develop controlled natural languages on top of the resource grammars without the application programmer knowing the mechanics of the resource grammars.

Grammatical Framework (GF henceforth) is a toolkit used for rapid development of multilingual grammar resources and applications based on the functional programming paradigm, the logic framework of abstract syntax plus concrete syntax. GF is also a grammar formalism grounded on categorical formalism [23] [24]. GF has one abstract syntax which defines categories of trees and the functions to implement them and many concrete syntaxes, one for each specific language grammar which provides the linearization of the categories and function of trees embodied in abstract grammar [22]. These parallel grammars of concrete syntaxes equivalent to parallel multiple context-free grammars reside in Grammar Resource Library (GRL) [25] [26] and currently, it has over 35 languages forming the multilingual ecosystem of GF [13] [27]. The GRL is divided into morphological and syntactic components [22]. In the morphology component, inflection smart paradigms are built using the regular expression [22] to build morphological lexicons of categories while in the syntax component, implementation of syntax rules is done. In GF, parsing transforms language-specific concrete syntax into abstract trees (language analysis) while linearization transforms abstract trees to strings in a specific language (language generation).

Grammar features are defined using parameters which are objects of some type and use the keyword *param.* Below is an illustration of parameter number

param

Number = Singular| Plural

GF makes a distinction between inherent and variable features of grammar. To gather all features of a specific category together, a record is used. For example, the noun category in Kikamba languages has inherent feature class gender and variable features number and case and therefore its linearization type record gathering all features would be defined as below

N = {*s*: *Number* => *Case* => *Str*; *g*: *Cgender*};

Finally, GF uses the operator "+" for concatenation and keyword *oper* to define operation or function for regular expression of all categories in the morphology modules.

4. Implementing the Kikamba Grammar in GF

Dictionaries, linguistic postgraduate theses and informants (who speak the language and/or are linguists) formed the data source for the lexicon and descriptive grammar. Linguists were used in cleaning, authenticating the data and through elicitation, they generated morphology and syntax of the categories that were missing in the Descriptive grammar from corpora. The elicitation was performed either through language analysis of the corpus through linguist judgment or by translation from English to the specific Bantu language as proposed

by Chelliah [28]. Snowball sampling techniques [29][1], which is a non-probabilistic sampling technique was used to gather the sparse corpora and to identify the few linguists available in the language. The evolutionary prototype model [30][2] approach was applied since for every function or module developed in GF there was a need to demonstrate its working by testing and refining the function until it produces the correct output. Interlingua rule based approach was used to develop the computational grammar in a morphology driven strategy, which is a bottom-up method. It involves first defining the lexicon, then categories, their smart paradigms based on the regular expression and finally working on the syntax rules [25]. Therefore, we will first discuss the morphology of the part of Speech tags and thereafter syntax rules.

4.1. Noun

To model the noun inflection class gender, number and case, grammar features were used. Ten class genders were identified as shown in **Table 1** column 1 and coded to ease their use as per column 3 since this work is a subset of a project to create a computational grammar for Bantu languages in Kenya. The case consisted of normative and locative. The locative case was created by adding suffix "ni" to normative case lexicon, while the number refers to singular and plural. The noun inflects number to the case with an inherent class gender feature as shown by the linearization categories of a noun (lincat) below.

```
param
Number = Sg | Pl;
Case = Nom | Loc;
Cgender = G1|G2 | G3 | G4 | G5 | G6 | G7| G8 | G9 | G10;
lincat
N = {s: Number => Case => Str; g: Cgender};
```

Kikamba language has a simple noun (single string) and compound noun (two strings) with inflection happening by changing the prefix (see **Table 3**). The smart paradigm regN implements the simple noun while compoundN implements the compound noun. The nouns that do not inflect were modeled using iregN (irregular nouns). The make noun mkN function assembled the smart paradigms together as shown below together with snippets of the smart paradigms.

```
mkN = overload {
mkN: Str -> Cgender -> N = \n, g -> lin N (regN n g);
mkN: (man, men: N)-> Cgender -> N = compoundN;
mkN: (man, men: Str) -> Cgender -> N = \s,p,g -> lin N (iregN s p g);};
```

The function **PrefixPlNom** provided the inflection prefix while each smart paradigm retained class gender for future concord agreement with the noun modifiers at the syntax stage.

[1]http://explorable.com/snowball-sampling
[2]http://www.softdevteam.com/Evolutionary-lifecycle.asp

Table 3. Kikamba noun morphology.

Noun inflection

regN: **Str ->Cgender -> Noun =** \w, g **-> let** wpl **= case** g **of {**
G1=>case w **of {**"mwa" **+ _ => Predef.**drop **2** w;
"mwi" **+ _ =>** "e" **+ Predef.**drop **3** w;
_ => PrefixPlNom G1 + Predef.drop **2** w **};**
G2=>case w **of {**"mw" **+ _ =>** "my" **+ Predef.**drop **2** w;
_ => PrefixPlNom G2 + Predef.drop **2** w **};**
……..
_ => PrefixPlNom g **+ Predef.**drop **2** w**};**
in iregN w wpl g;
 	
compoundN: **N -> N ->Cgender-> N =** \mundu**,**muume**,**g **-> {**
s **= **n**,**c **=>** mundu.s**!** n**!** c **++** muume.s**!**n**!** c;
g **=** g; lock_N **= <> };**
 	
iregN: **Str-> Str ->Cgender -> Noun=** \man**,**men**,**g **-> {**
s **=** table**{Sg =>** table**{Nom =>** man;
Loc=> man **+** "ni" **|** men **+** "ni" **};**
Pl => table**{Nom =>** men; **Loc=>** ""**}};** g **=** g**; };**

4.2. Adjective

Adjectives were implemented using parameter AForm, which had positive (AAdj), comparative (AComp) forms plus Adverbs9Advv) formed using adjectives and utilizing variable features: class gender and number. The comparative adjective form was implemented by adding the infix "ang" to positive adjective form just before the final vowel of the adjective. Table 4 provides a snippet of the smart paradigms for the regular adjective (regA). The function **ConsonantAdjprefix** provided the specific class gender prefix for the adjective. The concatenation of the class gender prefix with a vowel starting Adjective root was affected by morph phonological process.

AForm **=** AAdj Cgender Number **|** AComp Cgender Number **|** Advv;

4.3. Verbs

The GRL provided a grid of (4*2*2) four tense (present, past, future and conditional), two polarities (positive and negative) and two anteriorities (anterior and simultaneous) which were used to implement verbs. The above grid expanded because of morphemes in Kikamba verbs, which depend on ten class gender and number grammar features such as subject marker and object marker hence (10*2*4*2*2). To improve time and space complexity, we implemented the verb suffixes in Table 2 at the verb level and the prefixes to be concatenated at the verb phrase level.

Various verb forms needed for implementation of the verb and verb phrase were identified as present progressive, infinitives, past tense form, present definite form and neutral form and the parameter VForm was used to assemble them as shown below. The parameter VForm Extension provided the derivational morphology based on the extension suffixes presented in Table 2. The

Table 4. Kikamba adjective morphology.

Adjective inflection
regA:**Str -> {**s: **AForm => Str} =** \seo **-> {**s **=** table **{** **AAdj G1 Sg=>case Predef.**take **1** seo **of {** "a"**|**"e"**|**"i"**|**"o" **=>** "mw" **+** seo**;** "u" **=>** "m" **+** seo**;** **_ => ConsonantAdjprefix G1 Sg +** seo **};** **……** **AComp** g **Sg=>let** af: **Str = case Predef.**take **1** seo **of {** "i" **=>** "mw" **+** seo**;** "a" **=>** "my" **+** seo**;** "u" **=>** "m" **+** seo**;** **_ => ConsonantAdjprefix** g **Sg +** seo **};** **in** init af **+** "ang" **+** last af **} };**

smart paradigms regV and iregV were the functions for regular and irregular verbs.

```
param
VExte = EPassive | EApplicative | EReciprocal | ECausative | EDistributive;
VForm = VPreProg | VInf |VPast | VPreDef | VGen | VExtension VExte;
oper
regV: Str -> Verb =\vika -> let root = init vika
in {s = table{
VPreProg => case Predef.dp 1 root of {
"b" |"v"|"m" => root + "ete";
_ => root + "ite"};
VInf => "ku"+ vika;
VPast => root + "ie";
VPreDef => root + "aa";
VExtension type => init vika + extension type + last vika;
VNeuter => vika}};
iregV: Str -> Verb =\vika -> {s=\\_=> vika};
```

4.4. Numeral

Cardinal and ordinal numerals were implemented both in words from 0 up to 999,999. Two parameters were used to model numerals. First, DForm with four forms unit represents ranges of 0 - 9 numerals, tens representing a range of 10 - 99 and hund 100 - 999 range. The CardOrd represents ordinal (Nord) and cardinal (Ncard) numerals. The smart paradigm regular number (regNum) was used to implement the numerals. Ordinal numerals were formed from cardinal numerals by adding class gender morpheme supplied by function Ordprefix.

```
param
DForm = unit | teen | ten | hund;
CardOrd = NCard | Nord;
oper
regNum: Str -> {s: DForm => CardOrd => Cgender => Str} =
```

\six -> {s = table {
unit => table {NCard =>\\g => six;
NOrd => \\g => Ordprefix g ++ six};
teen => table {NCard =>\\g => "ikumi na" ++ six;
NOrd => \\g => Ordprefix g ++ "ikumi na" ++ six};
ten => table {NCard =>\\g => "miongo" ++ six;
NOrd => \\g => Ordprefix g ++ "miongo" ++ six};
hund => table {NCard =>\\g => "maana" ++ six;
NOrd => \\g => Ordprefix g ++ "maana" ++ six} } };

4.5. Personal Pronouns and Possessives

The personal pronoun is a string but requires concord agreement of class gender, number and person since GF treats it as a noun phrase while the possessive inflect by class gender and number. The PronForm parameter was used to represent the above two scenarios as depicted below with the function make pronoun mkPron generating both lexemes by taking two string, class gender, number and person as arguments being supplied by the linearization lin of the pronoun as shown by the example he_Pron below. Finally, the function ProunSgprefix and ProunSgprefix provided the class gender-specific prefix for concatenation with possessive form stem as shown in Figure 1.

param
Agr = Ag Cgender Number Person;
PronForm = Pers | Poss Number Cgender;
lin
he_Pron = mkPron "we" "ake" G1 Sg P3;

4.6. Other Morphology Categories

The demonstrative, quantifier and preposition configured as a string dependent on class gender and number parameters. Adverbs do not inflect hence are independent strings. The linearization type of preposition was configured with a Boolean operator to distinguish between the ones being fused with nouns and those not. Below are the linearization category and the smart paradigm mkprep.

lin
above_Prep = mkPrep "iulu" False;
oper
Prepp = {s: Number => Cgender => Str; isFused: Bool};

```
mkPron: (i, mine : Str) -> Cgender -> Number -> Person ->
   {s: PronForm => Str ; a : Agr} = \i,mine, g,n,p ->
        { s = table {
            Pers => i;
            Poss n g => case <n,g> of {
              <Sg ,_> => ProunSgprefix g + mine ;
```

Figure 1. Function for forming pronoun.

mkPrep = overload {

mkPrep: Str ->Bool-> Prep = \str,bool -> lin Prep {s = \\n,g => str; isFused = bool};

mkPrep: (Number => Cgender => Str) ->Bool-> Prep = \t,bool -> lin Prep {s = t; isFused = bool}; };

4.7. Common Noun (CN)

In Indo-Europeans languages, the CN is combined with an adjective to form NP or another CN and later a determiner can be added as a pre-modifier or post-modifier. However, in Kikamba language, the determiner is added between the adjective and the noun. Thus, the design of CN using two strings as exemplified below was to enable string one "s" to hold the CN while string two "s2" to hold the adjective. Hence it would be easier to add a determiner between string one and two. The class gender was retained from the noun since it will be used in agreement (concord). Below is the rule for forming CN from an adjective and a noun. All noun modifiers come after it with the exception of some quantifiers. Kikamba language does not have articles.

lincat

CN = CNoun;

oper

CNoun: Type = {s: Number => Case => Str; g: Cgender; s2: Number => Str};

CN has pre and postmodifiers such as an adjective, relative clause, adverbs, sentence and noun phrase and based on them, ten syntax rules were constructed. Below is an example of combining an adjective and a common noun.

AdjCN ap cn = {s = cn.s; g = cn.g; s2 = \\n => cn.s2! n ++ ap.s ! cn.g ! n};

4.8. Determiner Phrase (Det)

Det Phrases can either be possessive or demonstrated which were implemented using quantifiers, numbers and possessive pronouns. Three rules were implemented for Det Phrase and below is an example of one of the rules which form Det by taking a quantifier and a number.

DetQuant quant num = {s = \\Cgender =>quant.s ! num.n!Cgender ++ num.s !Cgender;

n = num.n; isPre = True};

4.9. Adjective Phrase

The adjective phrase was modeled via positive adjective, comparative adjective, post modifier of an adjective such as adverbs and also attaching it to a sentence. In total, eleven rules were used to implement adjective phrases and the comparative adjective phrase. The next rule exemplifies the implementation. The agreement consists of number and class gender and the Boolean value allows us to place the adjective phrase after the noun.

ComparA a np = {s = \\g,n => a.s !AAdj g n ++ "kuvita" ++ np.s ! npNom; isPre = False};

4.10. Noun Phrase (NP)

NP was implemented from the common noun, proper names, determiners, pronouns and also recursion of NP with adverbs, pre-determiners and determiners. NP implementation used two parameters: case and agreement (concord). On the case, we introduce extra case NPoss to cater for NP formed from personal and possessive pronouns. Eight rules were implemented to form NP. Below is an example of how to form NP by combining a determiner and a common noun in the Kikamba language. The Boolean function associated with the determiner allows pre and post determiners of CN to be placed in the right position.

```
DetCN det cn = {s =\\c=> case det.isPre of {
False => det.s!cn.g ++ cn.s ! det.n !npcase2case c    ++ cn.s2!det.n;
True => cn.s ! det.n !npcase2case c ++ det.s!cn.g ++ cn.s2!det.n};
a =Ag cn.g det.n P3;};
```

4.11. Verb Phrase (VP)

In VP the prefixes (focus, negation, subject marker, tense) morphemes were concatenated to verbs as mentioned in section 3.3 to make a complete verb. Since a whole verb can act as a sentence, then the parameters of sentences: polarity, tense and anterior in addition to agreements were used in the design as exemplified by the operation oper verb phrase. Five record strings were used: *s* for normal verb, *progV* for progressive verbs, *compl* for object of the verb, *imp* for imperative verbs and *inf* for infinitive verbs. The subcategorization of verbs was taken care of through *compl* (one place, two place and three place verb) and in total 20 rules were implemented based on the regular verb phrase function regVP.

```
oper
VerbPhrase: Type = {
s: Agr => Polarity => Tense => Anteriority => Str;
compl: Agr => Str;
progV: Str;
imp: Polarity => ImpForm => Str
inf: Str};
```

4.12. Other Syntax Categories

A clause was formed by combing a noun phrase and a verb phrase and implemented the topology SVO where the O was the second string of verb phrase which implemented the compliment of the verb. In the next section, we illustrate one of the rules for forming clauses. The clauses formed a sentence with the same parameters. However, the difference in GF is that the polarity and tense in clauses are undetermined [25]. Finally, the sentence and interrogative forms utterance (utt), which were the starting category for this computational grammar and was modeled based on definition 2. Seven clause rules, eight utterance rules and seven sentence rules were implemented

```
PredVP np vp = let agr = verbAgr np.a in{s=\\pol,tense,anter => let
verb: Str = vp.s!Ag agr.g agr.n agr.p !pol!tense!anter;
obj: Str = vp.compl !Ag agr.g agr.n agr.p; in
np.s !npNom ++ verb ++ obj};
```

5. Results

The Kikamba grammar was subjected to test suites for purposes of testing and evaluation. The testing aimed to improve grammar quality (reduce over the generation and ensure coverage) during development while the evaluation objective was to check coverage and quality of the grammar after development. The linguistic phenomena covered for this grammar are shown in **Figure A1** and are the ones that were tested and evaluated. There are three ways used to create test suites for testing computational grammars [31] [32].

- Grammar writer or expert writes the test suite data or uses already existing test suites.
- Using natural existing corpus or treebanks.
- Use of the comments created for each grammar rule that shows what the rule parses in the grammar.

We based our evaluation and testing on the aspect of the grammar already developed as per **Table 5**. Thus, we used method one for evaluating and method three for testing.

To create the test suite for testing, the comment(s) for each function/rule in the abstract syntax was used. The comments are/is an example(s) of what the rule can parse in the English language in addition to extra phrases generated by the grammar writer for each rule in the English language. The test suite for each rule was translated into Kikamba language phrases or lexicon (gold standard). The rule was implemented in such a way that its linearization output to match the gold standard, else the function was refined and the regression testing re-run until a match was obtained and also in case of changes of the module, re-runs were made to ensure no new noise was introduced. The above is the standard testing procedure for GF grammar [25] and also illustrated in **Figure 2**.

Table 5. Grammar coverage.

	Coverage
Sentence	Declarative, Questions
Tense	Present, Future, Past and Conditional
Verb	One-Place, Two-Place, Verb Phrase
Determiners	Quantifiers, Numbers and Possessive Pronoun
Noun	One Place Two-Place, Three Place Complex Noun
Adjective	Positive, Comparative and Complex
Noun Phrase	Personal Pronoun and NP Phrase
Adverb	Modifying Verbs, Numbers and Adjective
Others	Prepositional and Conjugation

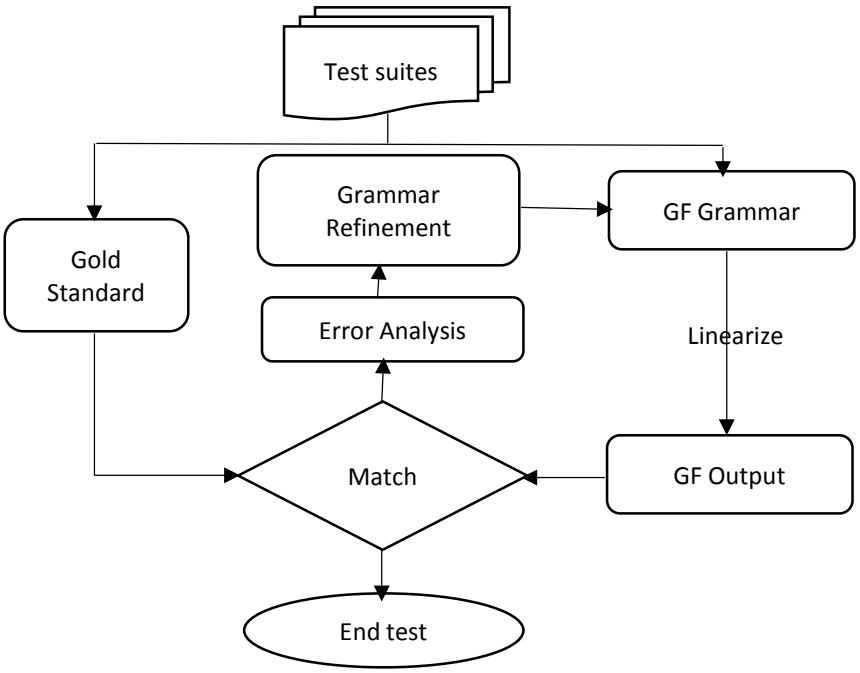


Figure 2. Testing process.

In our evaluation a 100 sentences test suite was developed from three sources: a linguist who was provided with the 500 different categories lexicons in GF so as to generate sentences, GF online treebanks[3] and Khegai [33] Russian work. The test suite was translated by a Kikamba language expert into the Kikamba test suite (the gold standard). Using the GF Kikamba grammar, the test suite was linearized into the Kikamba language. Where a sentence produced more than one linearization because of lexical variant or synonyms, then the one that best fit in reference to the gold standard was taken. The gold standard and the linearization output were matched using the online Tilde[4] machine translation platform and also the error rate Perl scripts in order to extract the metrics: Bilingual Evaluation Understudy (BLEU), Word Error Rate (WER) and Position Independent Error Rate (PER) which are commonly used metrics for evaluating machine translation [34]. BLEU [35] (ranges from 0 to 1 or expressed as a percentage) demonstrated a good correlation of machine translation to human judgment and PER and WER based on Levenshtein distance [36] were excellent metrics to investigate the errors in Kikamba language since it has a lot of nasal insertion, deletion and substitute. The results were: cumulative 4-gram BLEU of 83.05%, WER of 12.82% and PER of 10.96%.

We shall demonstrate how coverage of morphology and syntax using the dominate topology was accomplished in four levels. The Graphviz[5] software will be used to provide the Kikamba parse tree and words alignment after parsing the equivalent in English.

[3]https://github.com/GrammaticalFramework/gf-rgl/tree/master/treebanks
[4]https://www.letsmt.eu/Bleu.aspx
[5]http://www.graphviz.org/

- Normal sentence with simple SVO topology
- A sentence with a complex Noun Phrase
- Prepositional usage
- Normal questions and Wh-questions

Figure 3 represents the sentence "these bad men will cut many trees" in Kikamba languages. The verb "cut" is a two-place verb hence has an object existing in future tense with positive polarity and simultaneous anteriority. The Sentence S is created from the clause Cl, which consists of NP and VP. Also, the VP is made of VPslash and NP. Therefore, the sentence is indirectly made of NP VPslash NP, which represents the SVO structures respectively. Table 6 shows the morphology of individual categories.

Figure 4(a) and Figure 4(b) demonstrates a complex noun phrase and one place verb in word alignment and parse tree respectively; thus, no object in the sentence. The gloss of the sentence is "all your big brothers didn't sleep". The NP consists of Noun, possessive determiner, adjective and determiner and the tense of the sentences is past tense with negative polarity and simultaneous anteriority. The morphology is discussed in Table 7. All tense, polarity and anteriority implemented in this grammar have been exemplified in Table A1 at the appendix using the verb "sleep".

Figure 5(a) demonstrates the use of the auxiliary verb "to be", the preposition

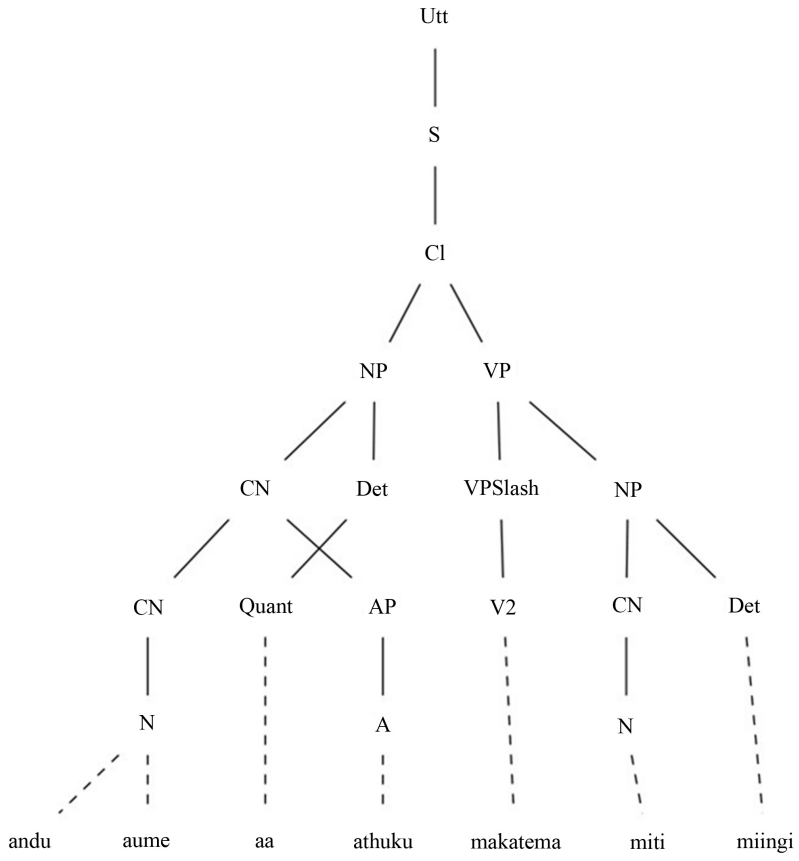
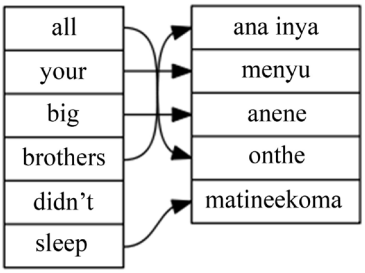


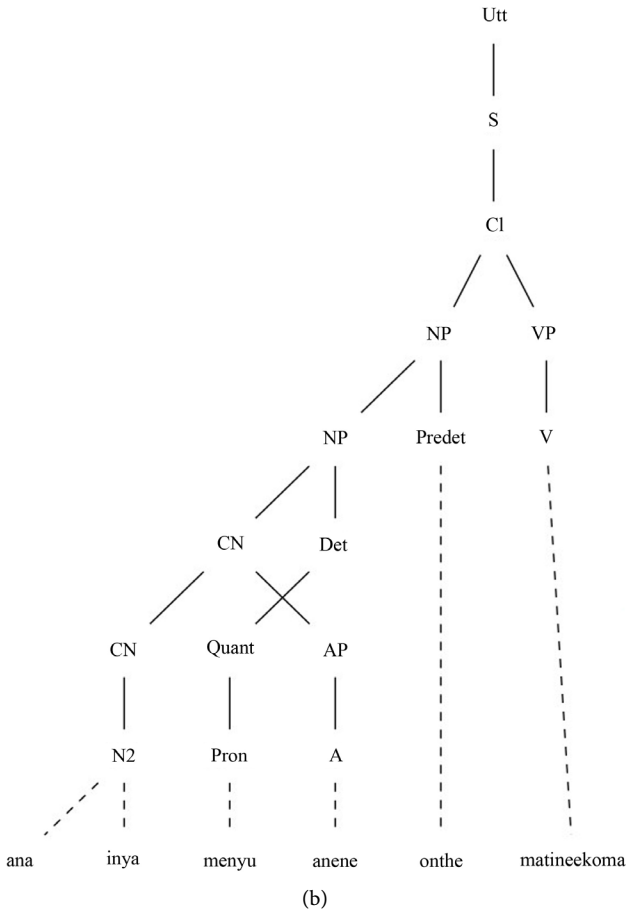
Figure 3. Utterance in Kikamba.

Table 6. Words morphology.

Word	Category	Explanation
Andu aume	Compound Noun	*a* class gender G1 number Pl prefix *ndu*-root *a* class gender G1 number Pl prefix *uume*-root
Aa	Quantifiers	class gender G1 dependent string
Athuku	Adjectives	*a* G1 concord prefix *thuku* Adj root
Ma	VP	Subject marker for class gender G1 and person 3
Ka	VP	Future tense morpheme in simultaneous
Tema	V2	Two place verb (with argument)
Miti	N	*mi* class gender G2 number Pl prefix *ti*-root
Miingi	Determiner	*mi* G1 concord prefix *ingi* Det root



(a)

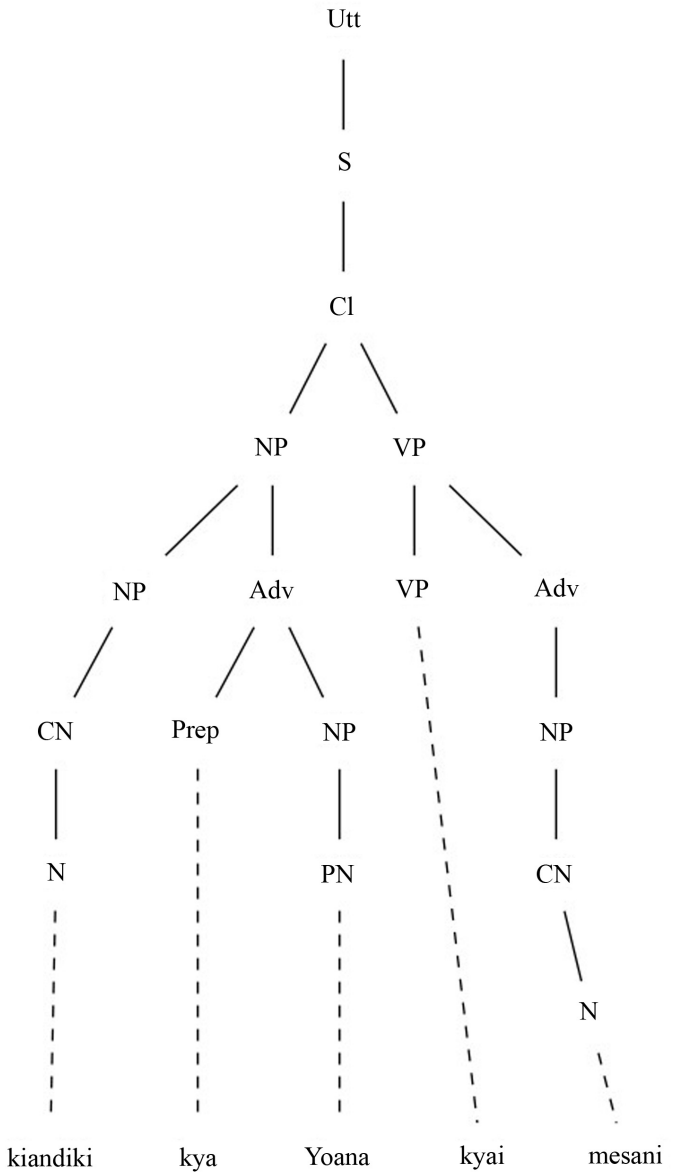


(b)

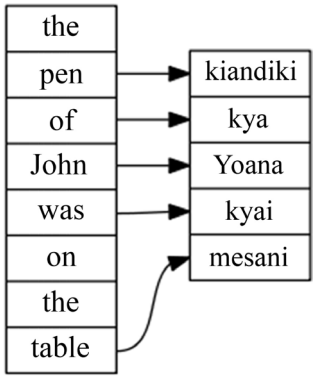
Figure 4. (a) Utterance in Kikamba; (b) Utterance in Kikamba.

Table 7. Words morphology.

Word	Category	Explanation
Ana Inya	N2	*Prefix a* class gender G1 number Pl *root ndu* *String to the noun*
menyu	Possessive Det	class gender G1 dependent string
Anene	Adjective	*a* G1 concord prefix and the adjective root is nene
Onthe	Determiner	class gender G1 dependent string
Ma	VP	Subject marker for class gender G1 and person 3
Ti	VP	past tense morpheme in simultaneous
nee		Infix
koma	V	*mi* class gender G2 number Pl prefix *ti*-root



(a)



(b)

Figure 5. (a) Preposition usage; (b) Word alignment.

"on" that is fused with the noun "table" to become "mesani" and preposition "of" which is translated "kya" based on class gender G4 of the pen. The gloss of the utterance used is "the pen of John was on the table". **Figure 5(b)** shows word alignment between English and Kikamba languages for the same utterance.

In Kikamba language, the tone is used to mark a question; hence, there are no rearrangements of the declarative sentence constituents. **Figure 6(a)** demonstrates the coverage of Wh-question "which trees did the wind push?" while the **Figure 6(b)** shows the word alignment of the Wh-question in English and Kikamba, while **Figure 7(a)** shows the yes-no question using the question "did the students play the song" and the word alignment are demonstrated in **Figure 7(b)**.

The Kikamba grammar is part and initial stage of creating a shared grammar for Kenyan Bantu languages through bootstrapping strategies, mainly grammar sharing and grammar porting. In order to maintain a standard regression testing of any new Bantu language that will be added via bootstrap, we parsed the hundred English sentences in order to create a treebank test suite. **Table 5** represents the categories covered in the treebanks. Below is an example of a tree which will linearize into "andu aume miongo ili athuku vyu nimananyw'ie nzovi" in the Kikamba language with a gloss of "the twenty very bad men drank beer" in the English language. The tree starts at the phrase level (PhrUtt) with no conjugation and vocative, taking sentence utterance (Utts). The clause (UseCl) is in the past tense, has a positive polarity and simultaneous anteriority. The function DetCN creates the noun phrase while ComplSlash creates the Verb phrase of a two-place verb drink with the function MassNP creating the compliment of the VP as a noun phrase as shown below.

PhrUtt NoPConj (UttS (UseCl (TTAnt TPast ASimul) PPos (PredVP (DetCN (DetQuant DefArt (NumCard (NumNumeral (num (pot2as3 (pot1as2 (pot1 n2))))))) (AdjCN (AdAP very_AdA (PositA bad_A)) (UseN man_N))) (Com-plSlash (SlashV2a drink_V2) (MassNP (UseN beer_N)))))) NoVoc

The treebanks created had 2854 functions in total. With the largest tree having 62 functions while the shortest had 11 functions. The largest tree was made of two sentences which had complex verb phrases and noun phrases.

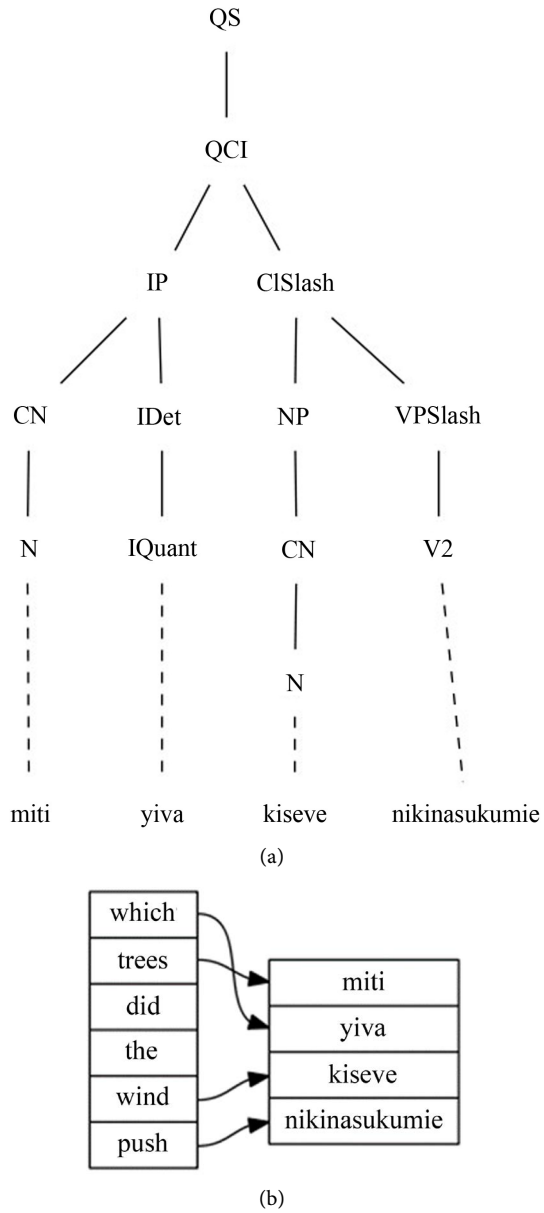


Figure 6. (a) Wh-question; (b) Word alignment.

6. Discussion

The statistical machine translation (SMT) Dholuo-English and Swahili-Dholuo [37] work gave a low BLEU score of 0.29 and 0.15, which the author attributed to lack of bilingual corpora. Given that the corpus was divided into ten portions; nine portions used for training and one portion used for testing, then the expectation was a high BLEU score. This is a clear indication that the use of a rule-based system will produce a high performance for under resourced languages. The SAWA corpus English to Swahili statistical machine translation [38] resulted in a BLEU score of 35, which is still low. Weku [39] reports a BLEU score of 32.6 on English-Swahili SMT based on Bayesian inference. We could not find a rule based system evaluation using the above metrics so as to compare

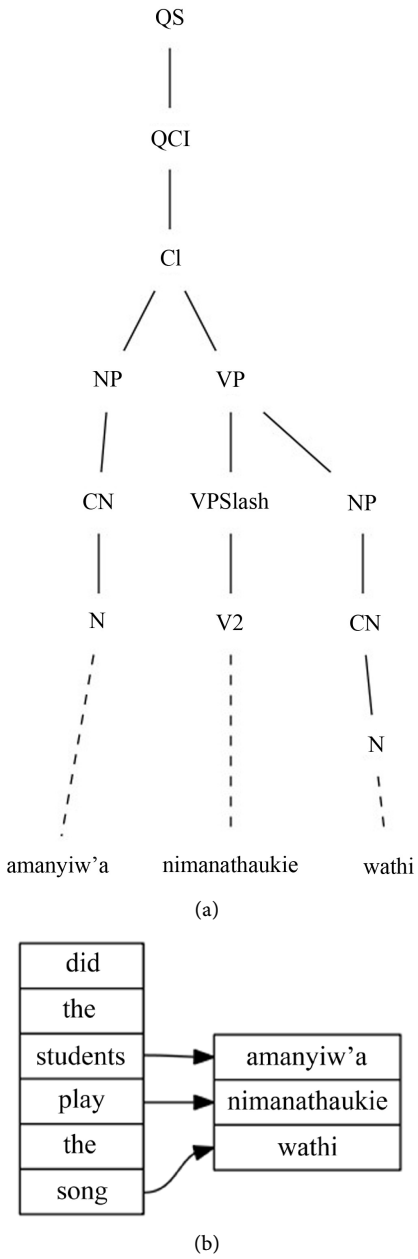


Figure 7. (a) Question; (b) Word alignment.

with and especially not a system for a Bantu language. Therefore, this work is a clear indication of how using the rule based system will help to produce highly accurate systems for these under resourced languages.

The error analysis was done sentence by sentence and **Figure 8** summaries the issue which contributed to the noise. In Kikamba language, pronouns were dropped (prop drop) since they were represented in the subject marker of the verb and in some cases, they were not dropped. Secondly, some prepositions were fused in the noun but also had strings. Verbs contributed the most significant percentage of the errors due to morphophonological issues as a result of nasal deletion and insertion, which is present in the Kikamba language [40].

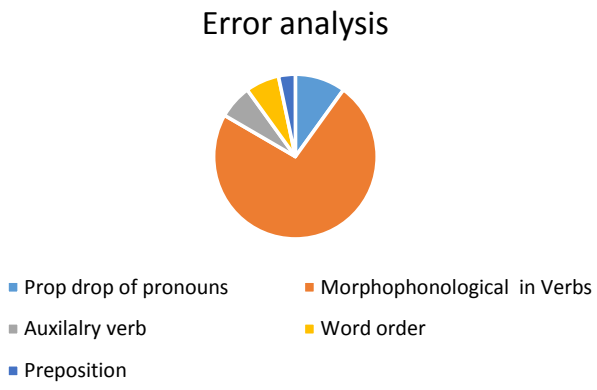


Figure 8. Preposition usage.

When a sentence had two adjectives, their order was changed in the translation and was heavily penalized by WER and BLEU hence the use of PER which allows words re-order and the error reduced to 10.96% from 12.82% of WER.

7. Conclusions

Through this paper, we have formalized the grammar for Kikamba language through the high precision rule-based approach in interlingua GF environment. The metrics results after evaluation which are encouraging are 4-gram BLEU of 83.05%, WER of 12.82% and PER of 10.96%. Therefore our contribution would be: firstly, we have provided NLP tools; morphological analyzer and machine translator for under-resourced Kikamba languages by extending the GF library, which is a step towards BLARK. Secondly, the wide coverage of the Kikamba computational grammar provides a platform for building multilingual technological applications and also to generate the scarce bilingual corpus pairing with other languages present in GF for experimenting using data driven methods. Finally, we have also created a treebank that can be used to evaluate Bantu languages.

Future work would be working on the morphophonological rules of verbs, extending the lexicon so as to handle text and finally including questions as part of the grammar.

Acknowledgements

We would like to acknowledge the contribution made by the following people in terms of Kikamba translation, Kikamba grammar structure, GF expertise. Prof. kyalo Wamitila, Prof. Angelina Kioko, Dr. Hans Leiß, Dr. Otiso Wambua, Obed Mutiso, Joe Kyalo, Christopher Kithuka, immaculate Wanza and Rama Munara.

Conflicts of Interest

The authors do not have any conflict of interest.

References

[1] Guthrie, M. (1948) The Classification of the Bantu Languages. The International

African Institute by the Oxford University Press, Oxford.

[2] Kaviti, L.K. (2004) A Minimalist Perspective of The Principles and Parameters in Kikamba Morpho-Syntax. Doctoral Dissertation, University of Nairobi, Kenya.

[3] Mbuvi, M.K. (2005) The Syntax of Kikamba Noun Modification. Unpublished Master's Dissertation, University of Nairobi, Kenya.

[4] Welmers, W.E. (1973) African Language Structures. University of California Press, Oakland, CA.

[5] Kioko, A.N., Njoroge, M.C. and Kuria, P.M. (2012) Harmonizing the Orthography of Gikuyu and Kikamba. In: Iribemwangi, P., Ogechi, O.N. and Odour, N., Eds., *Book Harmonization and Standardization of Kenyan Languages*, *Orthography and Other Aspects*, The Centre for Advanced Studies of African Society, Cape Town, South Africa, 39-63.

[6] Kioko, A.N. (1995) The Kikamba Multiple Applicative: A Problem for the Lexical Functional Grammar Analysis. *South African Journal of African Languages*, **15**, 210-216. https://doi.org/10.1080/02572117.1995.10587081

[7] Munyao, K.M. (2006) The Morph Syntax of Kikamba Verb Derivations: A Minimalist Approach. The University of Nairobi, Nairobi, Kenya.

[8] Roberts-Kohno, R.R. (2000) Kikamba Phonology and Morphology. Doctoral Dissertation, Ohio State University, Columbus, OH.

[9] Mutiga, J.M. (2002) The Tone System of Kikamba: A Case Study of Mwingi Dialect. Doctoral Dissertation, University of Nairobi, Kenya.

[10] Kituku, B., Musumba, G. and Wagacha, P. (2015) Kamba Part of Speech Tagger Using Memory-Based Approach. *International Journal on Natural Language Computing*, **4**, 43-53. https://doi.org/10.5121/ijnlc.2015.4204

[11] Kituku, B., Wagacha, P. and De Pauw, G. (2011) A Memory-Based Approach to Kıkamba Named Entity Recognition. *Proceedings of the Conference on Human Language Technology for Development*, Cairo, Egypt, 106-111.

[12] Ng'ang'a, W. (2012) Building Swahili Resource Grammars for the Grammatical Framework. Shall We Play the Festschrift Game? In: Santos, D., Lindén, K. and Ng'ang'a, W., Eds., *Shall We Play the Festschrift Game*? Springer, Berlin, Heidelberg, 215-226. https://doi.org/10.1007/978-3-642-30773-7_13

[13] Pretorius, L., Marais, L. and Berg, A.A. (2017) GF Miniature Resource Grammar for Tswana: Modelling the Proper Verb. *Language Resources and Evaluation*, **51**, 159-189. https://doi.org/10.1007/s10579-016-9341-z

[14] Krauwer, S. (2003) The Basic Language Resource Kit (BLARK) as the First Milestone for the Language Resources Roadmap. *Proceedings of SPECOM*, 8-15.

[15] Hyman, L.M. (1979) Phonology and Noun Structure. Aghem Grammatical Structure, 1-72.

[16] Kihm, A. (2002) What's in a Noun: Noun Classes, Gender and Nounness. Ms. Université Paris, Paris.

[17] Demuth, K. (2000) Bantu Noun Class Systems: Loan Word and Acquisition Evidence of Semantic Productivity. Classification Systems. In: Senft, G., Ed., *Book System of Nominal Classification*, Cambridge University Press, Cambridge, 270-92

[18] Ibrahim, M.H. (2014) Grammatical Gender: Its Origin and Development. Walter de Gruyter, **160**.

[19] Reichenbach, H. (1947) The Tenses of Verbs. Time: From Concept to Narrative Construct: A Reader.

[20] Rugemalira, J.M. (2007) The Structure of the Bantu Noun Phrase. *SOAS Working Papers in Linguistics*, **15**, 135-148.

[21] Kituku, B., Lawrence, M. and Wanjiku, N. (2016) A Review on Machine Translation Approaches. *Indonesian Journal of Electrical Engineering and Computer Science*, **1**, 182-190. https://doi.org/10.1090/psapm/012/9981

[22] Curry, H.B. (1961) Some Logical Aspects of Grammatical Structure. *Structure of language and Its Mathematical Aspects*, **12**, 56-68.

[23] Ranta, A. (2009) GF: A Multilingual Grammar Formalism. *Language and Linguistics Compass,* **3**, 1242-1265. https://doi.org/10.1111/j.1749-818X.2009.00155.x

[24] Paikens, P. and Gruzitis, N. (2012) An Implementation of a Latvian Resource Grammar in Grammatical Framework. *LREC* 2012, 1680-1685.

[25] Ranta, A. (2011) Grammatical Framework: Programming with Multilingual Grammars. CSLI Publications, Center for the Study of Language and Information, Stanford, CA.

[26] Ljunglöf, P. (2004) Expressivity and Complexity of the Grammatical Framework. Doctoral Dissertation, Chalmers University, Sweden.

[27] Ranta, A. (2006) Type Theory and Universal Grammar. Philosophia Scientiæ. Travaux d'histoire et de philosophie des sciences. 115-131. https://doi.org/10.4000/philosophiascientiae.415

[28] Chelliah, S.L. (2001) The Role of Text Collection and Elicitation in Linguistic Fieldwork. In: Newman. P. and Ratliff, R., Eds., *Book Linguistic Fieldwork*, Cambridge University Press, Cambridge, 152-165. https://doi.org/10.1017/CBO9780511810206.008

[29] Ngau, P. and Kumssa, L. (2004) Research Design, Data Collection and Analysis. Training Manual No. 12. United Nations Centre for Regional Development, Africa Office.

[30] Carr, M. and Verner, J. (1997) Prototyping and Software Development Approaches. Department of Information Systems, City University of Hong Kong, Hong Kong., 319-338.

[31] Bröker, N. (2000) The Use of Instrumentation in Grammar Engineering. *Proceedings of the* 18*th Conference on Computational Linguistics,* **1**, 118-124. https://doi.org/10.3115/990820.990838

[32] Butt, M. and Tracy, H.K. (2003) Grammar Writing, Testing and Evaluation. In: Ali, F., Ed., *Book Handbook for Language Engineers*, Stanford, CA, 129-180.

[33] Khegai, J. and Ranta, A. (2004) Building and Using a Russian Resource Grammar in GF. In: *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, Berlin, Heidelberg, 38-41. https://doi.org/10.1007/978-3-540-24630-5_4

[34] Vilar, D., Xu, J., Luis Fernando, D.H. and Ney, H. (2006) Error Analysis of Statistical Machine Translation Output. *LREC* 2006, Genoa, Italy, 697-702.

[35] Koehn, P. (2004) Statistical Significance Tests for Machine Translation Evaluation. *Proceedings of the* 2004 *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, July 2004, 388-395.

[36] Levenshtein, V.I. (1966) Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, **10**, 707-710.

[37] De Pauw, G., Maajabu. N. and Wagacha, P.W. (2010) A Knowledge-Light Approach to Luo Machine Translation and Part-of-Speech Tagging. In: *Proceedings of the*

Second Workshop on African Language Technology (*AfLaT* 2010), European Language Resources Association (ELRA), Valletta, Malta, 15-20.

[38] De Pauw, G., Wagacha. P.W. and de Schryver, G.M. (2011) Towards English-Swahili Machine Translation. Research Workshop of the Israel Science Foundation.

[39] Weku, O.V. (2014) Use of Bayesian Model for Word Alignment in Swahili-English Statistical Machine Translation. Master's Dissertation, University of Nairobi, Kenya.

[40] Kioko, A. (1999) The Verb 'Be' in Kikamba: Issues in Identifying the Form. *Chemchemi International Journal of Arts and Social Sciences*, 94-105.

Appendix A

Table A1. Examples of tense, negation and anteriority.

Form	Swahili	English
TPresASimulPPos	Nimakomaa	they sleeps
TPresASimulPNeg	we ndakomaa	he doesn't sleep
TPastASimulPPos	nimanakomie	they slept
TPastASimulPNeg	inyui mutineekoma	you didn't sleep
TFutASimulPPos	ithyit ukakoma	we will sleep
TFutASimulPNeg	we ndukakoma	you won't sleep
TCondASimulPPos	makeethiwa makomie	they would sleep
TCondASimulPNeg	maikeethiwa makoma	they wouldn't sleep
TPresAAnterPPos	ithyi nitwakoma	we have slept
TPresAAnterPNeg	ithyi tuinakoma	we haven't slept
TPastAAnterPPos	we niwakomete	he had slept
TPastAAnterPNeg	we ndwakomete	you hadn't slept
TFutAAnterPPos	nyie ngeethiwa ninakoma	they will have slept
TFutAAnterPNeg	makeethiwa matanakoma	they won't have slept
TCondAAnterPPos	we niwesaa kukoma	she would have slept
TCondAAnterPNeg	we ndesaa kukoma	she wouldn't have slept

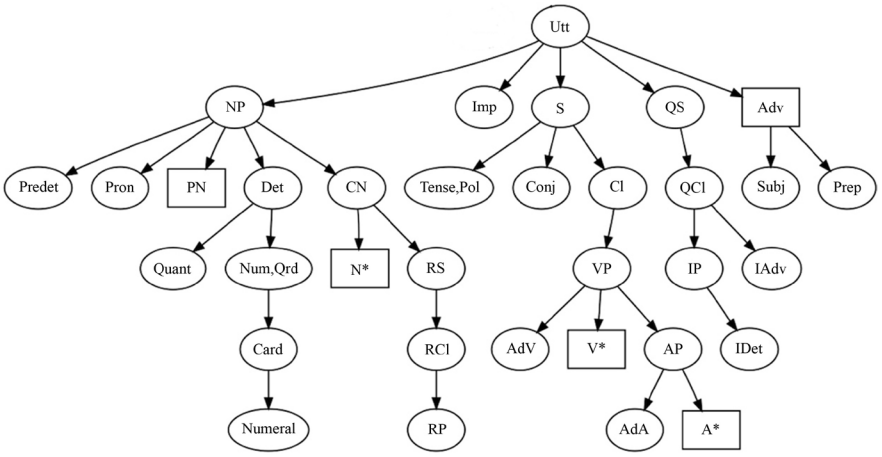


Figure A1. Treebank categories (adapted from [25]).